Bond-Graph Modeling

PETER J. GAWTHROP and GERAINT P. BEVAN

A TUTORIAL INTRODUCTION FOR CONTROL ENGINEERS

In the 19th century, Lord Kelvin and James Clerk Maxwell both observed that a wide range of phenomena give rise to similar forms of equations, finding analogies between heat flow and electric force and between lines of force and fluid streamlines. In the 1940s and 1950s, H.M. (Hank) Paynter of MIT worked on interdisciplinary engineering projects including hydroelectric plants, analog and digital computing, nonlinear dynamics, and control [1]. Through this experience, he observed that similar forms of equations are generated by dynamic systems in a wide variety of domain (for example electrical, fluid, and mechanical); in other words, such systems are analogous. Paynter incorporated the notion of an energy port into his methodology, and thus bond graphs were invented. Since that time, his group and many others have developed the basic concepts of bond-graph modeling into a mature methodology.

The bond-graph method is a graphical approach to modeling in which component energy ports are connected by bonds that specify the transfer of energy between system components. Power, the rate of energy transport between components, is the universal currency of physical systems.

The graphical nature of bond graphs separates the system structure from the equations, making bond graphs ideal for visualizing the essential characteristics of a system. Indeed, by creating bond graphs, designing and analyzing the structure of a system—perhaps the most important part of the modeling task—can often be undertaken using only a pencil and paper. Modelers can thus focus on the relationships among components and subsystems rather than the implementation details of their particular modeling software. Even before a computer is used, bond graphs can provide an engineer with information about constrained states, algebraic loops, and the benefits and consequences of potential approximations and simplifications.

Many computer-based modeling tools are available for generating and processing bond graphs, see "Further Reading." These tools generally have capabilities that extend far beyond those of traditional block-diagram software, including generation of symbolic representations, model inversion, and parametric identification as well as the ability to produce simulations, frequency responses, and other design aids. Bond-graph models can therefore be used by engineers not only to perform straightforward numerical analysis but also, more importantly, to gain qualitative insight.

1066-033X/07/$25.00©2007IEEE

This article has two purposes. First, we provide a tutorial introduction to bond graphs using examples that are familiar to control engineers. Second, the article is intended to motivate readers to apply the bond-graph approach themselves and to read more on the topic.

This article includes two case studies. The first case study is described in "Case Study 1: Laboratory Experiment," while the second is described in the section "Case Study 2: An Aircraft Fuel System." The early sections of this tutorial cover enough of the bond-graph method to model the system in Case Study 1. The later sections introduce more advanced topics required for Case Study 2.

Italic text (like this) highlights material that is directly applicable to the sidebar "Case Study 1."

ANALOGIES

Bond-graph modeling is based on three types of analogies, namely, signal analogies, component analogies, and connection analogies.

Signal Analogies

Table 1 shows four signal categories with examples from four engineering domains:

- **effort** signals, with the generic symbol e, including electrical voltage and mechanical force
- **flow** signals, with the generic symbol f, including electrical current and mechanical velocity
- **integrated effort** signals, with the generic symbol p, including electrical lines of flux and mechanical (translational and angular) momentum
- **integrated flow** signals, with the generic symbol q, including electrical charge and mechanical displacement.

Some entries in the table are less familiar but nevertheless provide a guide for the systematic choice of signals for system modeling. Additional domains, including magnetic and thermal, can also be incorporated in this scheme. A key insight is that the product of the effort and flow signals in each domain is power, that is,

$$\text{effort} \times \text{flow} = \text{power}. \quad (1)$$

For this reason, effort and flow signal pairs are deemed to be carried by the single power bond of Figure 1(a). The direction of the half-arrow indicates the positive direction of energy transport; in Figure 1(a), energy transport from left to right is regarded as positive.

The system shown in Figure S1 of Case Study 1 is driven by electrical power (voltage × current), which is converted to rotational mechanical power (torque × angular velocity) by a dc motor.

It should be noted that the half arrow does not denote an input or output in the same way as an arrow on a block diagram. Inputs and outputs are assigned by causal strokes, which are introduced later in the section "Causality and Block Diagrams."

Figure 1(b) shows the active bond, which carries either effort or flow. The active bond thus corresponds to a block-diagram signal and can therefore act as an interface between a system modeled as a bond graph and another system modeled as a block diagram.

Component Analogies

Table 2 lists one-port bond-graph components with analogous examples from four engineering domains. For example, the generic

- **Se** component, which can correspond to an ideal voltage source or an applied force, is a source of effort
- **Sf** component, which can correspond to an ideal current source or an applied velocity, is a source of flow
- **De** component, which can correspond to a voltmeter or a force sensor, is a detector of effort
- **Df** component, which can correspond to an ammeter or a tachometer, is a detector of flow
- **R** component, which can correspond to an electrical resistor or a mechanical damper, dissipates energy
- **C** component, which can correspond to an electrical capacitor or a mechanical spring (or compliance), stores energy
- **I** component, which can correspond to an electrical inductor or a mechanical mass, stores energy
- **SS** components (not shown in table) model colocated sensor-actuator pairs: **Se-Df** or **Sf-De**. These components also represent energy ports of compound components.

In the linear case, the corresponding equations for the **R**, **C**, and **I** components in terms of the generic variables of Table 1 are, respectively,

Further Reading

- Books: A. Mukherjee, R. Karmakar and A.K. Samantaray [23], D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg [3], P.J. Gawthrop and L.P.S. Smith [24], P.C. Breedveld and G. Dauphin-Tanguy [25], F.E. Cellier [26].
- Conference proceedings: [27]–[33].
- Journal special issues: [34]–[36].
- Web sites:
  - Bond Graph Compendium: www.ece.arizona.edu/~cellier/bg.html
  - Software reviews: www.bondgraphs.com/software.html
- Software:
  - Model Transformation Tools: mtt.sf.net
  - CAMP-G: www.bondgraph.com
  - 20-Sim: www.20sim.com
  - Dymola: www.dynasim.se
  - Symbols 2000: www.symbols2000.com

Case Study 1: A Laboratory Experiment

Figure S1(a) shows a laboratory experiment comprising a flexible beam driven by a dc motor through a gearbox. As with any physical system, it is up to the engineer to decide which effects to include and which to ignore. In this case, following the experimental manual, the main approximations are:

- The armature inductance of the dc motor is ignored.
- The beam is approximated by a rotational mass-spring system analogous to that of Figure 4(a).
- The gearbox is assumed to be rigid and free of backlash.
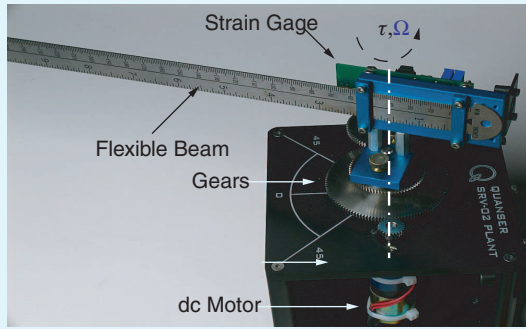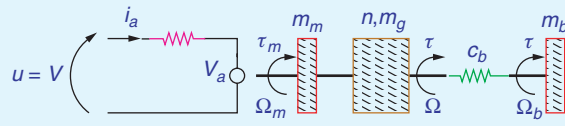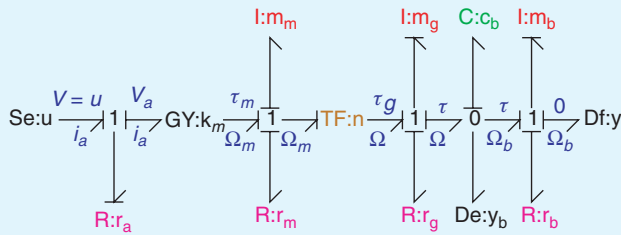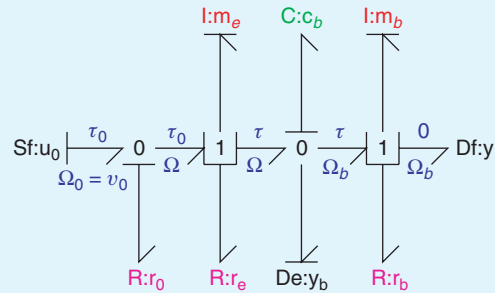
FIGURE S1 Case Study 1: a laboratory experiment. (a) The Quanser flexible beam experiment. The dc motor drives the flexible beam through a gearbox, a strain gauge measures the beam deflection, and a potentiometer measures the motor angular position. (b) System schematic diagram. For simplicity, the system is approximated as indicated in this mixed electrical/mechanical schematic diagram; in particular, the armature inductance is ignored, and the beam is given a lumped approximation. (c) The system bond graph. **R**:$\mathbf{r_a}$, **GY:k**, **I**:$\mathbf{m_m}$ and **R**:$\mathbf{r_m}$ model the motor; **TF:n**, **I**:$\mathbf{m_g}$, and **R**:$\mathbf{r_g}$ model the gearbox; and **C**:$\mathbf{c_b}$, **I**:$\mathbf{m_b}$, and **R**:$\mathbf{r_b}$ model the beam. **Sf:y** and **Se**:$\mathbf{y_b}$ measure Ω_b and τ_b, respectively. (d) A simplified bond graph. The drive components are combined into the four equivalent components **I**:$\mathbf{m_e}$, **R**:$\mathbf{r_e}$, **R**:$\mathbf{r_0}$, and **Se**:$\mathbf{u_0}$.

$$\mathbf{R}\{e = rf, \quad (2)$$

$$\mathbf{C}\begin{cases} e = \frac{q}{c}, & (3) \\ \dot{q} = f, & (4) \end{cases}$$

$$\mathbf{I}\begin{cases} f = \frac{p}{m}, & (5) \\ \dot{p} = e, & (6) \end{cases}$$

where r, c, and m are constants describing the corresponding physical system. In the electrical domain, (2) corresponds to Ohm's law and (3) to Coulomb's law; in the mechanical domain, (3) corresponds to Hooke's law, while (5) corresponds to Newton's second law.

Because the same type of component usually occurs more than once in a given system, the colon ":" notation is used to distinguish between multiple instances of each component type. In particular, the symbol preceding the colon refers to the component type, while the symbol following the colon labels the particular instance. Thus **C**:$\mathbf{c_1}$ refers to a **C** component labeled $\mathbf{c_1}$, which is equivalent to placing the label $\mathbf{c_1}$ adjacent to the symbol for a capacitor in an electrical circuit diagram.

Connection Analogies

Two components can be connected by a power bond thus giving them the same effort and flow. Figure 2(a) shows two mechanical components, while Figure 2(b) shows two analogous electrical components. Each of these physical systems can be represented by the bond graph of Figure 2(c).

Figure S1(b) shows the corresponding schematic. The left-hand part shows the conventional electrical model of the motor armature comprising the armature resistance r_a driven by the applied voltage V and the back EMF V_a. Since V is the system input, it is labeled by the conventional symbol u. The connection between the electrical and mechanical systems is given by the usual dc motor equations

$$\tau_m = k_m i_a,$$
$$V_a = k_m \Omega_m.$$

Figure S1(c) gives the bond graph corresponding to this apparatus.

Using the simplification rules of Figure 9, Figure S1(c) can be simplified to give Figure S1(d), where the **GY**, **TF**, and one of the **I** components have been removed. In terms of the original components, the components of the simplified system are

$$r_0 = \frac{k_m^2}{n^2 r_a},$$
$$r_e = r_g + \frac{r_m}{n^2},$$
$$m_e = m_g + \frac{m_m}{n^2},$$
$$u_0 = \frac{n}{k_m} u.$$

Although the simplified bond graph retains the same input-output dynamics as the original system, it is easier to understand. For example, it is clear that $\mathbf{R}:\mathbf{r_0}$ acts as a natural derivative controller as in Figure 11.

Figure S2 gives the frequency response relating the two outputs $y = \Omega_b$ and $y_b = \tau_b$ to the equivalent system input u_0. The frequency response, which is generated automatically from the bond graph and the numerical values of Table S1, is the first step in the control-design process.
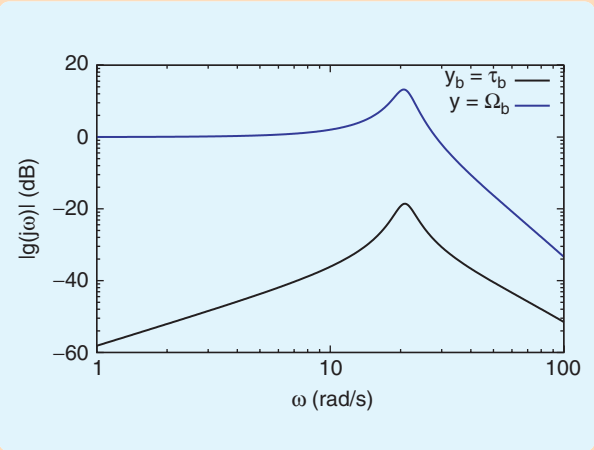
FIGURE S2 Frequency response for the Quanser experiment. The numerical values of Table S1 combined with the bond graph of Figure S1 yield the frequency response of $y = \Omega_b$ and $y_b = \tau_b$, measured in rad/s and N-m, respectively, to the equivalent system input u_0, measured in rad/s; the resonant peak corresponds to the flexure natural frequency of about 20 rad/s.

TABLE S1 Numerical values. These numerical values corresponding to each bond-graph component in Figure S1 are taken from the Quanser manual [14].

Component	Value	Units
k_m	0.00767	N-m/A
r_a	2.6	Ω
m_m	$3.87e^{-7}$	Kg-m^2
r_m	0.0	N-m-s/rad
n	$\frac{1}{70}$	
m_g	$2.5200e^{-5}$	Kg-m^2
r_g	0.0	N-m-s/rad
c_b	2	rad/N-m
m_b	0.0012368	Kg-m^2

TABLE 1 Analogous signals. Systematic modeling, including the bond-graph approach, uses the concept of analogous signals to bring together different physical domains. One such analogy is the effort/flow analogy displayed here, where each row contains analogous signals and each column corresponds to a domain. In each case, effort × flow = power.

Bond Graph	Translation	Rotation	Electrical	Hydraulic
Effort e	Force F N	Torque τ N-m	Voltage V V	Pressure P Pa
Flow f	Velocity v m/s	Angular velocity Ω rad/s	Current I A	Flow Q m^3/s
Integrated effort $p = \int e dt$	Momentum p kg-m/s	Angular momentum h kg-m^2/s	Lines of flux λ V-s	Momentum per unit area p kg-m/s
Integrated flow $q = \int f dt$	Position x m	Angle θ rad	Charge q C	Volume V m^3

Because there are only two components, the electrical components of Figure 2(b) share both voltage and current. But, more generally, electrical connections are either parallel [Figure 3(a)] or series [Figure 3(b)].

Figure S1 of Case Study 1 includes examples of both mechanical (damper) and electrical (resistor) R components.

The parallel connection obeys Kirchhoff's voltage law, whereas the series connection obeys Kirchhoff's current law. The bond-graph approach uses a **0** junction to model a parallel electrical connection [Figure 3(c)] and a **1** junction [Figure 3(d)] to model a series electrical connection. However, the series/parallel analogy can be misleading in the mechanical and other non-electrical domains; a more useful abstraction is to view an electrical parallel connection as a common-effort or **0** connection and an electrical series connection as a common-flow or **1** connection. The effort on each bond impinging on a **0** junction is equal, while the flows on these bonds sum to zero. A **0** junction with m bonds in and n bonds out is therefore described by the equation pair

$$\mathbf{0}\begin{cases} e_1^{\text{in}} = \cdots = e_m^{\text{in}} = e_1^{\text{out}} = \cdots = e_n^{\text{out}}, & (7)\\ \sum_{i=1}^{m} f_i^{\text{in}} - \sum_{j=1}^{n} f_j^{\text{out}} = 0, & (8)\end{cases}$$

where the efforts $e_1^{\text{in}}, \ldots, e_m^{\text{in}}$ and the flows $f_1^{\text{in}}, \ldots, f_n^{\text{in}}$ are carried on bonds pointing into the junction, while the efforts $e_1^{\text{out}}, \ldots, e_n^{\text{out}}$ and the flows $f_1^{\text{out}}, \ldots, f_n^{\text{out}}$ are carried on bonds pointing out of the junction. Likewise, the efforts on a **1** junction sum to zero while the flows are all equal, so a dual junction is described by the equation pair

$$\mathbf{1}\begin{cases} f_1^{\text{in}} = \cdots = f_m^{\text{in}} = f_1^{\text{out}} = \cdots = f_n^{\text{out}}, & (9)\\ \sum_{i=1}^{m} e_i^{\text{in}} - \sum_{j=1}^{n} e_j^{\text{out}} = 0. & (10)\end{cases}$$

The system shown in Figure S1 of Case Study 1 contains an electrical **1** *junction carrying armature current as well as several mechanical* **1** *junctions carrying velocity.*

The symbols **0** and **1** are chosen to be neutral with respect to the physical domain.

FIGURE 1 Types of bond. (a) A power bond. The *effort* and *flow* signal pair of Table 1 are carried by a single power bond. The half arrow indicates the direction of positive power transport. The signals at opposite ends are equal, that is, $e_2 = e_1$ and $f_2 = f_1$. Note that effort $\times$ flow $= e_1 f_1 = e_2 f_2 =$ power. (b) An active bond that carries either effort or flow. The active bond, which corresponds to a block-diagram signal, can act as an interface between a system modeled as a bond graph and another system modeled as a block diagram.

Simple RCI System

Figure 4(a)–(c) shows analogous systems drawn from three physical domains. Since the systems are analogous, they share the bond graph of Figure 4(d). As can be seen from Figure 4(a) and (c), the concepts of parallel and series connection can be misleading, whereas the concepts of common effort and flow junctions provide a domain-neutral formulation.

Power Conversion with Transformers and Gyrators

The effort and flow variables within each physical domain of Table 1 have different units and therefore cannot be

TABLE 2 Analogous components with one energy port. The analogous signals of Table 1 lead to the analogous components of this table; the first column gives the generic bond-graph component, while the remaining columns give the domain-specific analogues.

Bond Graph	Translation	Rotation	Electrical	Hydraulic
External				
Se	Applied force	Applied torque	Applied voltage	Applied pressure
De	Force sensor	Torque sensor	Voltmeter	Pressure sensor
	F N	T N/m	V V	P Pa
Sf	Applied velocity	Applied rotation	Applied current	Applied flow
Df	Speedometer	Tachometer	Ammeter	Flow meter
	v m/s	ω rad/s	i A	Q m^3/s
1 port				
C	Spring	Torsional spring	Capacitor	Accumulator
	K N/m	K N-m/rad	C F	K Pa/m^3
I	Mass	Moment of inertia	Inductor	Flow inertia
	m kg	J kg-m^2	L H	I kg/m^4
R	Damper	Rot. Damper	Resistor	Restrictor
	d N-s/m	d N-m-s/rad	R Ω	K Pa-s/m^3

directly connected. However, since power is the universal currency of physical systems, the power-converting bond-graph components **TF** (a generic transformer) and **GY** (a generic gyrator) of Figure 5(c) and (d) provide a means for converting power and thus connecting different domains. The **TF** component generalizes an electrical transformer, which has the property that the ratio of voltages (efforts) at the two terminals is the inverse of the ratio of current, which is consistent with the fact that power is conserved in the sense that the instantaneous power at the input port equals the instantaneous power at the output port at each instant of time. Figure 5(a) shows a physical system that, in idealized form, corresponds to the **TF** component of Figure 5(c). Additional examples of physical components with an ideal **TF** representation include a piston for mechanical-to-hydraulic power conversion and a rack-and-pinion gear converting translational to rotational power.

The **GY** component is the same as the **TF** component insofar as power is conserved; the difference is that flow at one port depends on effort at the other, and vice versa. Figure 5(b) shows a physical system that, in idealized form, corresponds to the **GY** component of Figure 5(d). The name gyrator arises from the property of a gyroscope that angular velocity (flow) is converted into torque (effort).

In the linear case, the **TF** and **GY** components have the equations

$$\mathbf{TF}\begin{cases} e_2 = ne_2, & (11)\\ f_1 = nf_2, & (12)\end{cases}$$

$$\mathbf{GY}\begin{cases} e_2 = kf_1, & (13)\\ e_1 = kf_2, & (14)\end{cases}$$

where n and k are nondimensional constants describing the corresponding physical system. The pairs (11)–(12) and (13)–(14) both describe energy-conserving components since, in both cases, the input and output power is the same, that is, $e_2 f_2 = e_1 f_1$.

The system shown in Figure S1 of Case Study 1 has a **GY** *component representing conversion of electrical to mechanical power within a dc motor as well as a* **TF** *component (a gear ratio) representing conversion of mechanical power within a gearbox. In each case, losses are accounted for using* **R** *components, which dissipate power.*
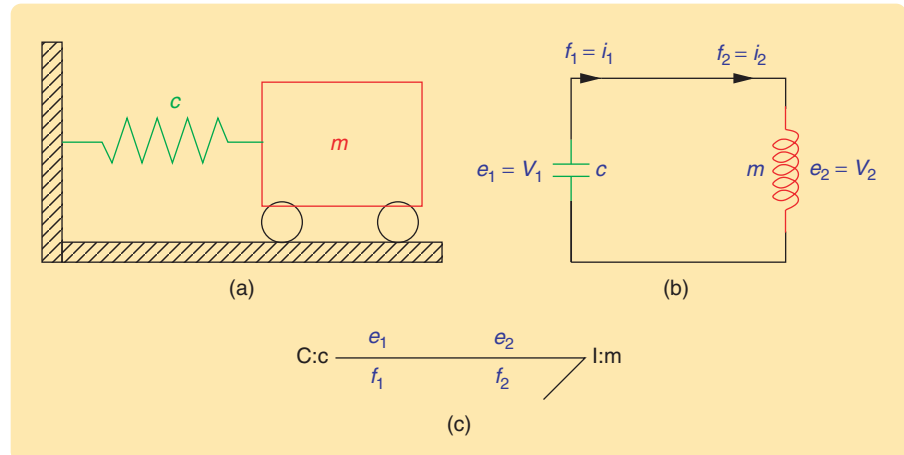
FIGURE 2 Connecting two components. (a) A mechanical mass m and spring with compliance c are connected together; the mass and spring share the same velocity (flow) $v_2 = v_1$ and the same action and reaction (effort) $F_1 = F_2$. (b) An electrical inductor with inductance m and capacitor with capacitance c are connected together; the components share the same current (flow) $i_2 = i_1$ and voltage (effort) $V_1 = V_2$. (c) The bond graph **I:m** and **C:c** components are connected together using the power bond of Figure 1(a); these components share a flow $f_2 = f_1$ and effort $e_1 = e_2$. The colon notation **I:m** and **C:c** associates the label **m** with the **I** component and the label **c** with the **C** component. The color coding is used to help interpretation; it is not part of the bond-graph method.
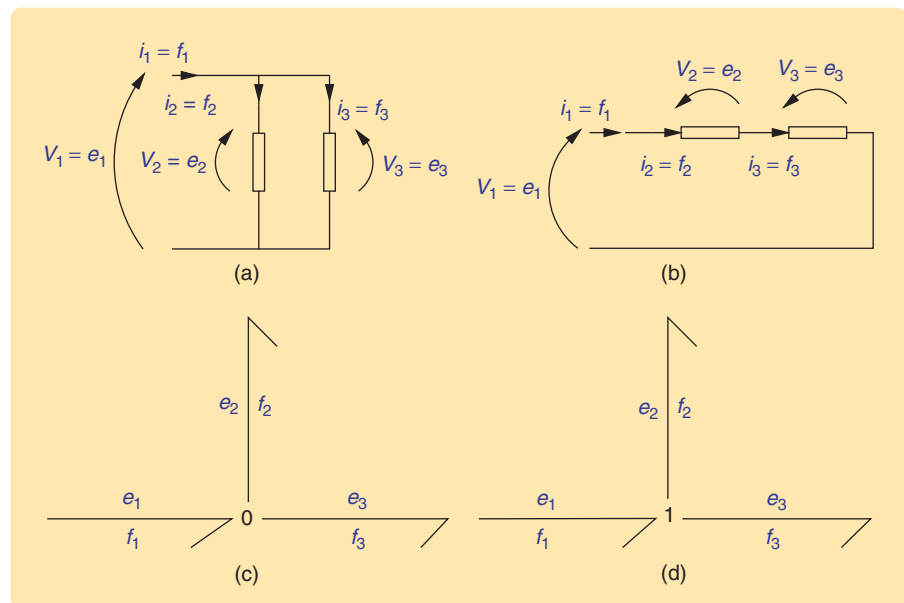
FIGURE 3 Connecting three components. (a) A parallel connection. This connection obeys Kirchhoff's voltage law; the voltage (effort) is common. (b) A series connection. This connection obeys Kirchhoff's current law; the current (flow) is common (c) Bond-graph generalization of diagram (a). In a **0**, or *common-effort*, junction, the efforts are equal and the flows sum to zero, that is, $e_1 = e_2 = e_3$, $f_1 - f_2 - f_3 = 0$. (d) Bond-graph generalization of diagram (b). In a **1**, or *common-flow*, junction, the flows are equal and the efforts sum to zero, that is, $f_1 = f_2 = f_3$, $e_1 - e_2 - e_3 = 0$.
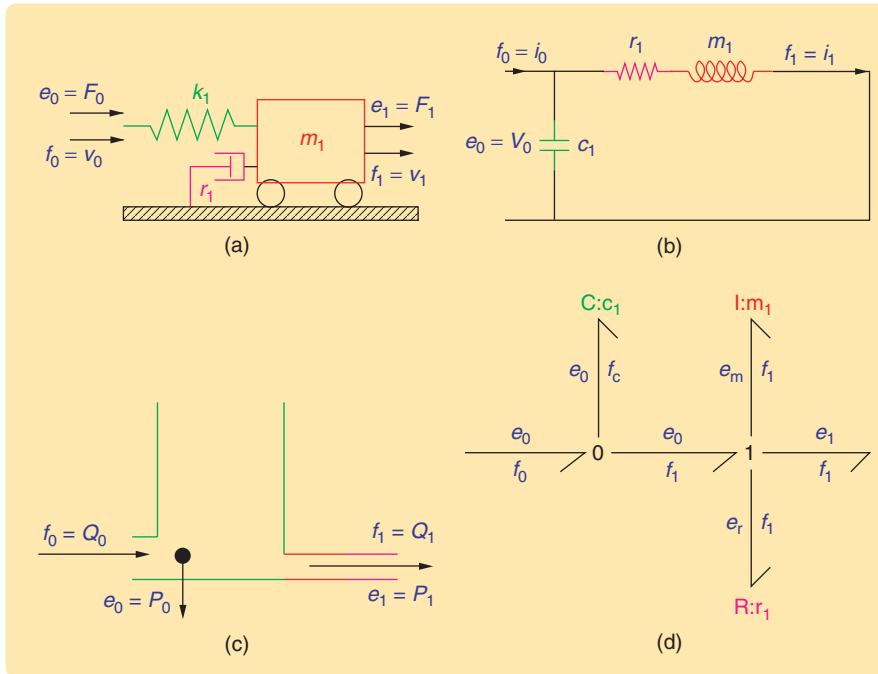
FIGURE 4 Analogous second-order systems. (a) Mechanical system comprising a trolley labeled m_1, a spring labeled k_1, and a damper labeled r_1. (b) Analogous electrical system comprising an electrical inductor labeled m_1, a capacitor labeled c_1, and a resistor labeled r_1. (c) Analogous hydraulic system, comprising a tank with an outlet pipe that has both resistive and inertial characteristics, corresponding to a flow-dependent pressure drop and the fluid momentum. (d) Bond graph representing the mechanical, electrical, and hydraulic systems. Each system has two energy ports, one at the left and one at the right, each corresponding to the domain-specific effort/flow pair of Table 1. Analogous components are the same color, the blue italic text indicating the signals in each case. Component **C** : $\mathbf{c_1}$ models extension of the spring, accumulation of charge in the capacitor, or storage of fluid in the tank. It would typically be parameterized by the spring stiffness k in the mechanical domain, by capacitance C in the electrical domain, and by tank cross-sectional area A and fluid density ρ in the hydraulic domain. Component **I** : $\mathbf{m_1}$ models the momentum of the trolley, the lines of the flux in the inductor, or the momentum of fluid in the pipe. It would typically be parameterized by the trolley mass m, electrical inductance L, or by the pipe length l and density ρ of the fluid within it. Component **R** : $\mathbf{r_1}$ models the friction of the damper, the resistance of the resistor, or the friction within the pipe. It is typically parameterized by a damping coefficient b, by an electrical resistance R, or by a hydraulic loss coefficient C_D and orifice cross-sectional area A.

CAUSALITY AND BLOCK DIAGRAMS

Although block diagrams are familiar to control engineers, the sidebar "Why Bond Graphs Are Better than Block Diagrams" explains that block diagrams have an unfortunate drawback, namely, they represent assignment statements rather than equations. In other words, a block diagram cannot be drawn until the inputs and outputs of each component are specified. For example, the upper right-hand part of Figure 6(a) shows an electrical resistor corresponding to the equation $V = ri$. Two possible block diagrams are shown below this component, where one has voltage (effort) output and corresponds to the assignment statement $V := ri$, while the other has current (flow) output and corresponds to the assignment statement $i := (1/r)V$. In contrast, the bond-graph representation in the upper left-hand part of Figure 6(a) is acausal and represents an equation. The addition of a causal stroke in each of the two lower bond graphs assigns the input and output of each **R** component. This causal assignment is not part of the initial modeling but is added later. This approach has the important advantage that bond-graph components are reusable within different
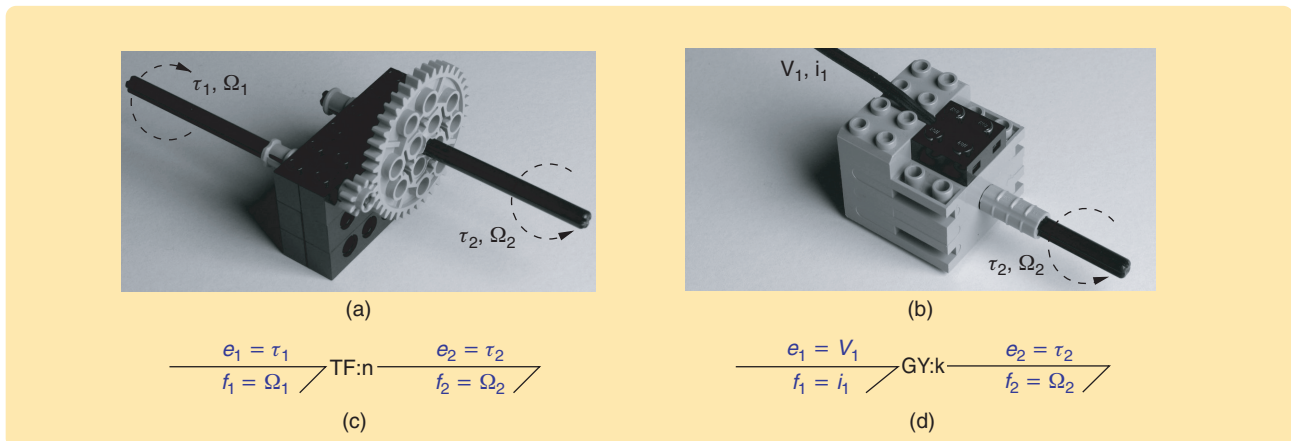
FIGURE 5 Transformers and gyrators. (a) A simple gearbox. Rotational mechanical power is converted between two shafts with gear ratio $n > 1$; the device corresponds to the *transformer* (**TF**) bond graph of (c). (b) a dc motor with power conversion between electrical terminals and the mechanical shaft, corresponding to the *gyrator* (**GY**) bond graph of (d). (c) The **TF** component provides power conversion such that $e_1 = ne_2$ and $f_1 = nf_2$. (d) The **GY** component provides power conversion such that $e_2 = kf_1$ and $e_1 = kf_2$, where k is the back EMF constant of the motor.

causal contexts, whereas block-diagram components are not. While the bond-graph approach thus has one model for a resistor, the block-diagram approach has two. In bond-graph terminology, the middle **R** imposes effort and has flow imposed on it, whereas the lower **R** imposes flow and has effort imposed on it.

Because the **C** component of Figure 6(b) is dynamic, the distinction between the two forms of causality is more significant. In particular, the middle diagram corresponds to integral causality, while the lower corresponds to derivative causality. The former is preferred if we wish to have a state-space system representation. The **I** component is similar to the **C** component of Figure 6(b) but with e and f reversed.

Figure 7(a) is the causally complete equivalent of Figure 4(d), Figure 7(b) is the corresponding block diagram, and the following comments explain the details:

- » The **C** and **I** components are in preferred integral causality; for the **C** component, this relation implies effort out and flow in, whereas, for the **I** component, this relation implies flow out and effort in.
- » The **R** component has effort output since the corresponding flow is caused by the **I** component.
- » The **0** junction has exactly one bond imposing effort on it, whereas the **1** junction has exactly one bond imposing flow on it.
- » The **0** junction of Figure 7(a) corresponds to the first summation block of Figure 7(b) and the connection to the second summation block.
- » The **1** junction of Figure 7(a) corresponds to the second summation block of Figure 7(b) and the various connections involving f_1.

Why Bond Graphs Are Better Than Block Diagrams

- » Acausal
  - Equation based
  - Causality, that is component input and output, determined *after* modeling
  - Causality issues clear
- » Energy conserving
  - Bonds convey *power*
  - Automatically obeys laws of physics
- » Compact
  - Each bond conveys *two* related signals
  - Connections are localized
  - Components are localized
  - Topology is closer to the physical system
  - Graphical depiction of sign convention
- » Reusable subsystems
  - Subsystem causality adapts in response to impinging subsystems
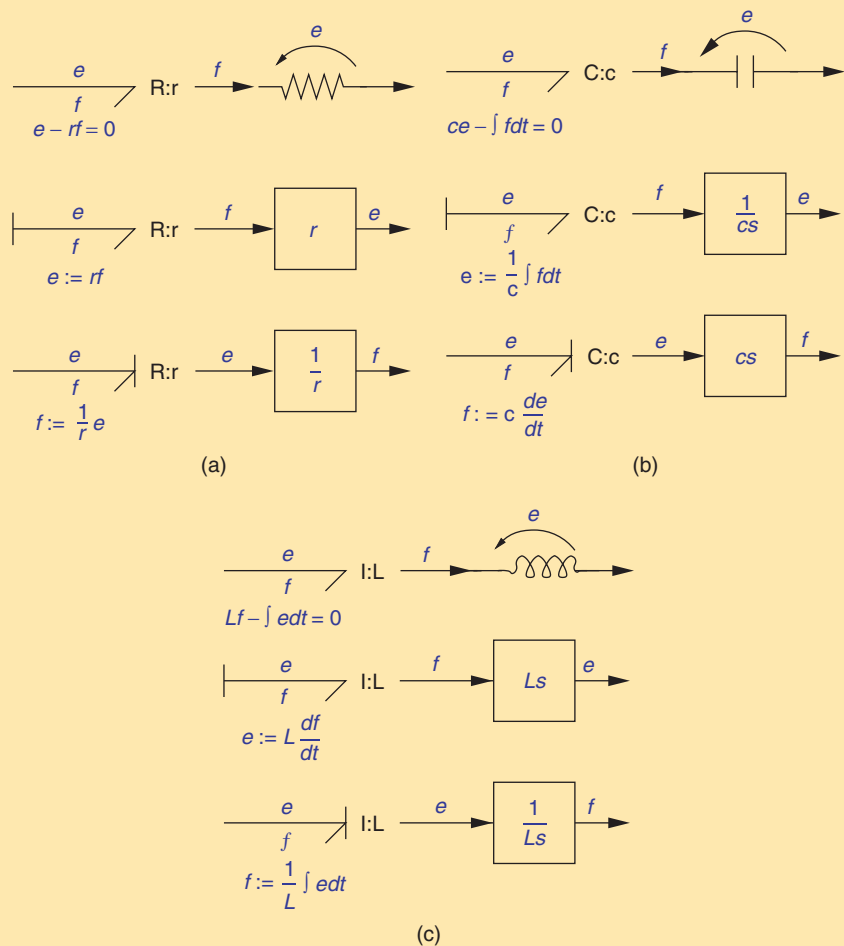
FIGURE 6 Component causality. (a) **R**. The topmost bond is *acausal* and represents an *equation*, for which the corresponding electrical component is shown. The middle bond has a causal stroke (perpendicular end-bar) indicating that e is the output, while the lower bond has a causal stroke indicating that f is the output. Because of the causal stroke, these two bonds represent *assignment statements* rather than equations. The corresponding block diagrams for these assignment statements are shown. (b) **C**. This component is similar to the **R** component except that the equation is differential, not algebraic. The middle diagram shows *integral causality*, while the lower diagram shows *derivative causality*. (c) **I**. This component is similar to the **C** component but with e and f reversed. The middle diagram shows derivative causality, while the lower diagram shows integral causality.

Causal Assignment

Abstracting the physical system as an acausal bond graph provides a complete description of the corresponding model. However, for the purposes of analysis, there are many ways of representing the system as a set of equations, and each such representation has its own particular uses. For example, the control engineer typically uses a state-space representation for system analysis and simulation, whereas the mechanical engineer may prefer a Lagrangian representation and the mathematician may prefer a Hamiltonian representation. Each of these representations corresponds to a particular causality and thus each representation can be extracted by imposing a particular pattern of causal strokes on the acausal bond graph [2], a procedure known as completing causality. This article focuses on generating a state-space representation of a system.

Figure S1 of Case Study 1 has examples of **R** *components in each type of causality. For example, the armature resistor* $\mathbf{R}:\mathbf{r_a}$ *imposes the armature current (flow), whereas* $\mathbf{R}:\mathbf{r_g}$ *and* $\mathbf{R}:\mathbf{r_b}$ *each impose torque (effort). Figure S1 also has examples of both integral (*$\mathbf{I}:\mathbf{m_g}$, $\mathbf{C}:\mathbf{c_b}$, *and* $\mathbf{I}:\mathbf{m_b}$*) causality as well as derivative (*$\mathbf{I}:\mathbf{m_m}$*) causality.*

The assignment of causality to a bond graph can usually be accomplished automatically by computer if the causality is specified at key points on the graph, usually the external ports, and if some general preference for integral or derivative causality [Figure 6(b) and (c)] is expressed by the modeler. The best known method for automating causal assignment is the sequential causal assignment procedure (SCAP) [3], which gives a state-space system representation. If, indeed, the system has a state-space representation, the details of the resulting pattern of causal strokes, although helpful in understanding the inner workings of the model, need not be viewed by the modeler. However, if the model does not possess a state-space representation, then the pattern of causal strokes clarifies the situation and helps the modeler reconsider the model in terms of the underlying physical system.

For proper causal completion, which will result in a set of explicit assignment statements, it is necessary that exactly one bond impose a flow on each **1** junction. Similarly, exactly one bond must impose an effort on each **0** junction. The causality of **TF** and **GY** components is also subject to constraints if self-consistent system models are to be generated. In particular, causality is transmitted unaltered through **TF** components, that is, one impinging bond imposes effort (flow), while the other has effort (flow) imposed on it. Causality is reversed through a **GY** component so that both impinging bonds impose effort (or flow) and have flow (or effort) imposed on them. Within these constraints, causality can be assigned arbitrarily, although general guidelines, or preferences, are usually expressed.

After specifying the causality at the external interfaces, it is generally advisable for the modeler to specify the preferred causality of the system **C** and **I** components. As discussed in Figure 6, **C** and **I** components may have either integral or derivative causality. For simulation or state-space representations, integral causality is usually preferable since it leads to ordinary differential equations (ODEs), which can be computed without recourse to computationally intensive differential algebraic equation (DAE) solvers.

Constraints due to **0** and **1** junctions or **TF** and **GY** components may make it impossible to place all of the energy storage elements in integral causality. In this case, dependent states result in DAEs.

Insofar as modeling is the art of approximation, that is, deciding which components, features, and behaviors to omit, bond graphs can help engineers decide which approximations are useful before generating the equations. For example, approximate models with derivative causality can be converted to integral causality either by adding greater detail to the model, in the form of additional states, or by combining states to simplify the model. An example of this conversion is the decision to model a shaft connecting two rotating masses as either rigid or compliant. The modeler may wish to consider whether the additional difficulty
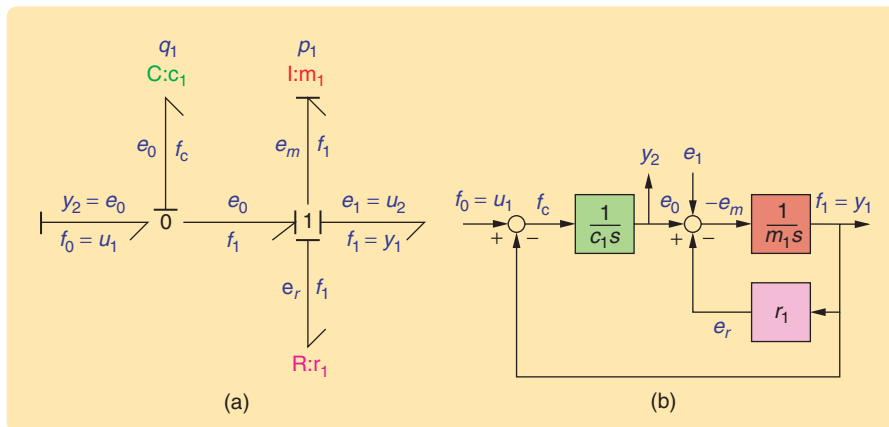
FIGURE 7 System causality. (a) Causal bond graph with causality completed on each component. As indicated by the causal strokes, the $\mathbf{C}:\mathbf{c_1}$ component has its preferred causality of flow in and effort out, the $\mathbf{I}:\mathbf{m_1}$ component has its preferred causality of effort in and flow out, whereas the $\mathbf{R}:\mathbf{r_1}$ component has, in this system, a causality of effort out and flow in. Annotations such as q_1 are for clarity and are not part of the bond-graph notation. (b) Block-diagram. The input and output of each component corresponds to the causal stroke on the bond-graph components of (a); the **C** and **I** components are in *integral* causality since they lead to block-diagram integrators. From Table 1, the associated states are the integrated flow q_1 and integrated effort p_1. The block diagram and bond graph share the same color coding to indicate the corresponding elements.

of solving DAEs outweighs potential disadvantages associated with an ODE representation, such as numerical stiffness or reduced transparency in the meaning of quantities represented by system states.

Although either causal configuration can be meaningful for an **R** component, the modeler should be aware of the possibility of algebraic loops arising between multiple **R** components. An algebraic loop occurs when the output of one component is required to determine the input of another, and vice versa. The formation of an algebraic loop exists on a bond graph if, between **R** components with opposing causality, there exists a path that includes neither a **1** junction with flow imposed by an **I** or **Sf** component nor a **0** junction with the effort imposed by a **C** or **Se** component. In other words, to avoid the formation of an algebraic loop, at least one system input or storage element with integral causality must impose a flow on a **1** junction or an effort on a **0** junction in each path between **R** components with opposing causality. A slight complication is that **GY** components effectively reverse the causality of the two subsystems on either side of them with respect to each other. Thus **R** components may have opposing causality even if the causal stroke is at the same end of the bond attached to each component due to a an odd number of **GY** components appearing in the path between them.

In Figure S1 of Case Study 1 the component $\mathbf{I}:\mathbf{m_m}$ *has derivative causality. An ordinary differential equation representation can be obtained by adding further detail, such as motor shaft compliance, or by simplifying the model.*

One of the benefits of bond-graph modeling is that the presence of algebraic loops is explicitly visible to the modeler. Although algebraic loops lead to implicit equations, such loops can be broken in various ways to allow the creation of an explicit model, for example:

- adjacent **R** components can be combined to form equivalent components
- additional dynamics can be modeled by introducing **C** or **I** components between the **R** components
- algebraic or numerical solvers can be introduced (using the **SS** component) to resolve the causal conflict
- the causality of adjacent components can be altered so that both **R** components can be assigned the same causality.

Despite the possibility of completely automating causal assignment, it is generally advisable for the modeler to be involved in the process since the causal assignments made within a model carry important information about the system and the suitability of the model for any intended purpose. In particular, the information gleaned from causal assignment gives the modeler immediate feedback as to the consequences of including or deleting a component from the bond-graph model of the physical system without the need to generate equations.

State-Space Equations and Block Diagrams

Bond graphs are an acausal system representation. By assigning a causal stroke to each bond, a causal representation can be generated. The causally complete model can be converted into other causal representations such as state-space equations and block diagrams. This section demonstrates the principles of this conversion. Most bond-graph software supports this conversion and provides an interface to standard control engineering tools such as Matlab and Octave [4].

The causal strokes on a bond graph provide "signposts" to guide the generation of state-space equations and block diagrams. Although software can perform these transformations automatically, state-space equations can be generated by hand.

Both the bond graph and simplified bond graph of Figure S1 of Case Study 1 correspond to state-space equations with three states since three components have integral causality in each case. The state-space equations can be derived by hand or by using symbolic software.

The following steps demonstrate this procedure using Figure 7(a):

- **Identify the states.** The system states are the integrated flows q associated with **C** components, as well as integrated efforts p associated with **I** components, in integral causality [Figure 6(b)]. In this case the states are q_1, the integrated flow variable associated with $\mathbf{C}:\mathbf{c_1}$, and p_1, the integrated effort variable associated with $\mathbf{I}:\mathbf{m_1}$.
- **Write state derivatives in terms of states and inputs.** By definition, $(d_{q_1}/dt) = f_c$. Following the causal strokes, $f_c = f_0 - f_1$. In this example, f_0 is an input and, following the causal strokes, $f_1 = (p_1/m_1)$, where p_1 is a state. Thus, the first state equation is

$$\frac{d_{q_1}}{dt} = f_0 - \frac{p_1}{m_1}. \tag{15}$$

By definition, $(d_{p_1}/dt) = e_m$. Following the causal strokes, we obtain $e_m = e_0 - e_r - e_1$. In this example, $e_0 = (q_1/c_1)$, and e_1 is an input. Using the properties of the **R** component given in Figure 6(a), $e_r = r_1 f_1 = r_1(p_1/m_1)$. Thus the second state equation is

$$\frac{d_{p_1}}{dt} = \frac{q_1}{c_1} - r_1\frac{p_1}{m_1} - e_1. \tag{16}$$

- **Write outputs in terms of states and inputs.** The outputs f_1 and e_0 can be written in terms of the states as $f_1 = (p_1/m_1)$ and $e_0 = (q_1/c_1)$. Defining

$$x = \begin{bmatrix} q_1 \\ p_1 \end{bmatrix}, \quad y = \begin{bmatrix} f_1 \\ e_0 \end{bmatrix}, \quad u = \begin{bmatrix} f_0 \\ e_1 \end{bmatrix}, \tag{17}$$

(15) and (16) can be written in state-space form

$$\frac{dx}{dt}(t) = Ax(t) + Bu(t), \tag{18}$$

$$y(t) = Cx(t), \tag{19}$$

The bond-graph method is an approach to modeling in which component energy ports are connected by bonds that specify the transfer of energy between system components.

where

$$A = \begin{bmatrix} 0 & -\frac{1}{m_1} \\ \frac{1}{c_1} & -\frac{r_1}{m_1} \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad C = \begin{bmatrix} \frac{1}{c_1} & 0 \\ 0 & \frac{1}{m_1} \end{bmatrix}. \quad (20)$$

In a similar fashion the block diagram of Figure 7(b) can be derived from the causally complete bond graph of Figure 7(a).

Case Study 1 uses simplification to reduce the number of bond-graph components and remove the component $\mathbf{I}:\mathbf{m_m}$ *in derivative causality.*

Simplification and Approximation

In the same way that two resistors in series can be combined into a single resistor or two rigidly connected masses can be combined into a single mass, bond-graph components can be combined to give a simplified bond graph with the same external behavior as the original bond graph. Such simplification can be useful in understanding the behavior of a complex system in terms of its simplified version.

Figure 8 shows two **I** components separated by a **C** component, a bond graph that can represent two rotating masses separated by a compliant shaft. Under steady state conditions, the masses rotate at the same speed, and there is no change in the twist of the shaft. This synchronous rotation is manifested as zero flow through the bond to the **C** component. Even under transient conditions, the flow is generally small. It may therefore be reasonable to approximate the flow to zero, an approximation that can be accomplished explicitly by replacing **C** with a flow source **Sf** adding zero flow to the junction. Having made this approximation, the bond graph can be simplified by eliminating the **Sf** component entirely because bonds adding zero flow to a **0** junction have no effect on the system.

Using some simple rules, further simplifications can easily be made. A junction connecting only two bonds is redundant since it merely constrains the effort and flow in each bond to be the same. An identical effect can be achieved by replacing both bonds and the junction with a single bond, thereby eliminating the **0** junction. Simplifying further, two junctions of the same type, which are connected by a single bond, can be replaced by a single junction of the same type. The result is a greatly simplified bond graph with two **I** components connected to a single **1** junction. Finally, the approximate system can be simplified as in Figure 9(a) and (b).

In more complex examples, analysis of the causality of the original model and the simplified version yields information that can be used to determine whether the approximation is effective. The original model has three energy storage states to which integral causality can be simultaneously assigned. The model therefore produces a set of three ODEs. The approximated model has only two states and is therefore in some sense simpler. However, following the rules of causal assignment, described previously, it is not possible to simultaneously place both of the **I** components in integral causality. One state must be placed in derivative causality [Figure 6(b)], which (without simplification) results in a set of DAEs. Whether the replacement of three ODEs with two DAEs is a useful approximation or not depends on the purpose of the model; nevertheless, a benefit of bond-graph causality is that the consequences of such an approximation are explicit to the modeler.
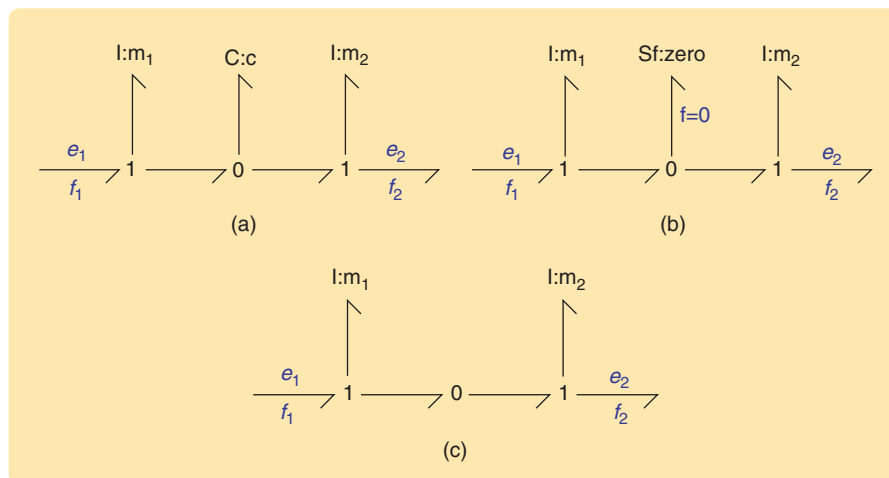
FIGURE 8 Approximation of bond graphs. (a) Three energy storage states. This bond graph might represent two rotating masses $\mathbf{I}:\mathbf{m_1}$ and $\mathbf{I}:\mathbf{m_2}$ connected by a flexible shaft **C:c**. (b) If the shaft has very low compliance, the shaft can be explicitly modeled as rigid by replacing the **C** component with a zero-flow **Sf**. (c) Since the addition of zero flow to a **0** junction does not affect the system dynamics, the bond graph can be simplified by removing the **Sf** component entirely.

ADVANTAGES OF BOND GRAPHS OVER BLOCK DIAGRAMS

Case Study 1 provides evidence for the assertions in the sidebar "Why Bond Graphs Are Better Than Block Diagrams." The evidence is presented below. In particular, Figure 10 is more complex than the bond graph of Figure S1 of Case Study 1.

The acausal, equation-based nature of a bond graph is apparent from the way that components can be treated identically regardless of the causality imposed on them. The resistances r_a and r_m of Case Study 1 are implemented identically in the bond-graph model of the motor with only the later addition of a causal stroke, specifying the inputs and outputs required to generate assignment statements from the acausal equations. By contrast, the same resistances r_a and r_m depicted in block-diagram form in Figure 10(b) are implemented differently, with the reciprocal of the resistance multiplying the input signal in one case, but not the other.

A more striking example can be seen in the handling of the motor inertia. Whether the inertia must be modeled with an integrator or differentiator changes according to its causality, as can be seen by comparing Figure 10(b) and (c), where a change to the input and output of the component model causes the inertia causality to change. In the first configuration, an output torque is generated in response to motor speed for a given voltage, and the inertia is represented by a differentiator.

The block diagram of Figure 10(a), corresponding to the bond graph of Figure S1 of Case Study 1 is complicated, bearing little relation to the system topology. A comparison with the bond graph reveals many of the benefits that bond graphs provide for modeling the system.

In the second configuration, the motor torque is an input, and the shaft speed an output, but the differentiator must then be changed to an integrator. It is also necessary to reverse the direction of some arrows and to change the sign of the signal connecting the motor torque τ_m to the motor mass, which is not necessary in the bond graph because the positive direction of energy transport is automatically handled by the sign convention specified by the bond directions.

The disadvantages of requiring two block diagrams to represent one component depending on its inputs and outputs extend beyond the need to create and maintain twice as many models. There is also the problem of unit testing. While it is a simple matter to verify the dynamic behavior of the block diagram with integral causality in Figure 10(c) by specifying constant inputs, this simplicity is not true of the version with a differentiator in Figure 10(c). Applying a constant shaft speed does not reveal any of the dynamic characteristics of the motor. The need for different forms of verification means that it is necessary to write separate tests for every particular instantiation of a model in block-diagram form,
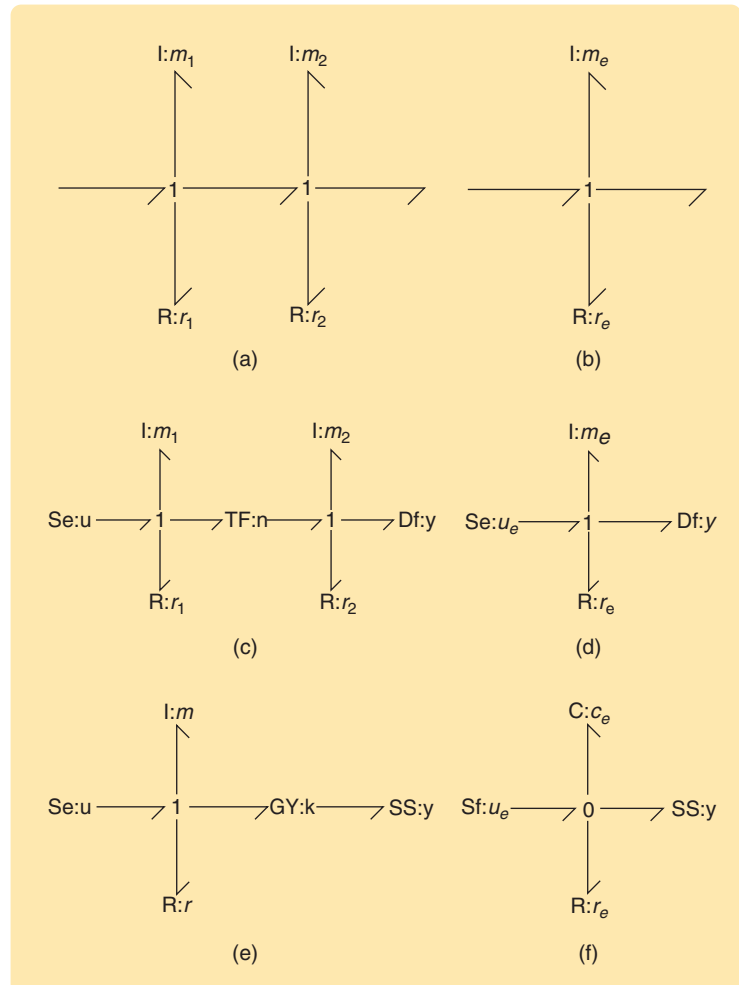
FIGURE 9 Simplification of bond graphs. (a) Connected **I** and **R** pairs. This bond graph might represent two rigidly connected masses, each of which is connected to a damper. (b) Simplified version of diagram (a). The parameters of each **I** and **R** component can combine as $r_e = r_1 + r_2$ and $m_e = m_1 + m_2$ to create a simpler model with equivalent properties. (c) **TF** connected **I** and **R** pairs. This bond graph might represent an electromechanical actuator in which electrical power is provided to a first-order electrical circuit, with inductance m_1 and resistance r_1, after which the power is transformed into mechanical power and applied to a mass-damper with parameters m_2 and r_2. (d) Simplified version of diagram (c). The system parameters can combine to form equivalent parameters $r_e = r_1 + n^2 r_2$ and $m_e = m_1 + n^2 m_2$. The interpretation of the input changes to $u_e = ku$ in accordance with the elimination of the gain, but the output remains the same. (e) Subsystem with **GY** connection. This bond graph can represent a linear electric pump with inductance m and resistance r. (f) Simplified version of diagram (e). Eliminating the **GY:k** component, the causality of the system reverses with respect to the output **SS:y**, requiring the **1** junction to be replaced by a **0** junction, and the **I:m** component to be replaced with the **C**:$\mathbf{c_e}$ component. The parameters associated with the system components also change to accommodate the elimination of the **GY** gain, giving $c_e = m/k^2$ and $r_e = k^2/r$. Again, the output remains the same, but the interpretation of the input changes to $u_e = ku$.

whereas it suffices to test a single configuration of a bond graph with confidence that the same model can be used regardless of the external causality.

Maintenance of bond-graph models is also made easier by the localization of components. It can be seen that the block-diagram representation of the gearbox requires that the gear ratio n be specified in two places in Figure 10(d), whereas the corresponding **TF**:**n** component need only be inserted once for the bond-graph representation (Figure S1). Multiple specification of parameters is usually only a minor inconvenience when initially creating models, but can easily lead to the introduction of hard-to-detect bugs when models are updated to accommodate future changes.

ADVANCED TOPICS

The bond-graph method has been developed in several ways since its inception, and many of these developments have been initiated at the biennial International Conference on Bond Graph Modeling. From these extensions, we discuss several topics that we believe to be of importance to control engineers. Sources of information for these topics can be found in "Further Reading."

Physical-Model-Based Control

As discussed in [5], [6] physical-model-based control regards feedback controllers as physical systems. This approach has the advantage that both physical intuition and energy-based stability analysis can be used to design
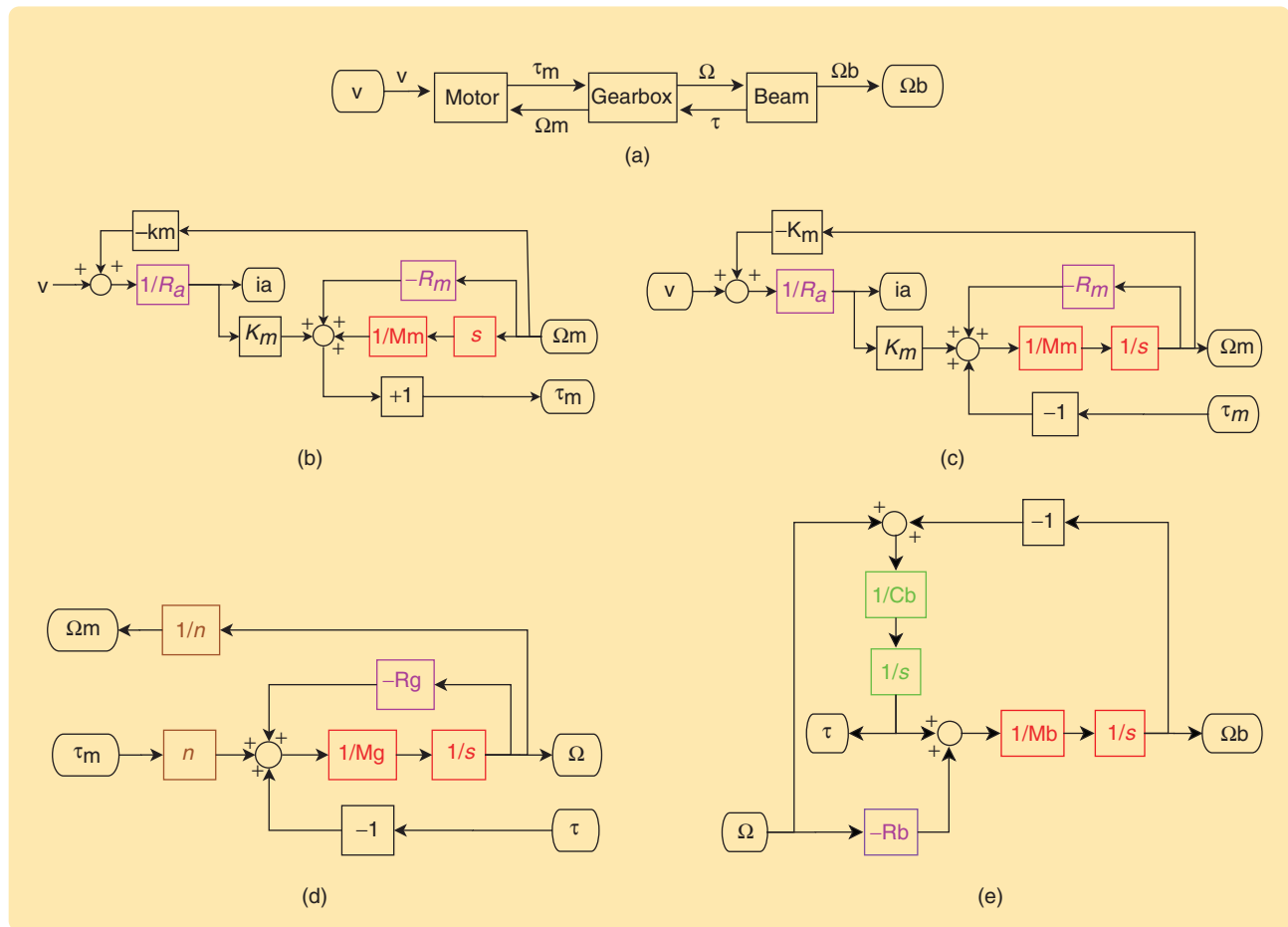
FIGURE 10 Block-diagram representation of the system described in Figure S1. (a) The top level system comprises three subsystems, namely, a motor, a gearbox, and a flexible beam. (b) The motor calculates the torque output in response to a given input voltage and shaft speed. Note that the resistances R_a and R_m (purple) are implemented differently; R_m appears as a multiplier, whereas the *reciprocal* of R_a multiplies its input. This difference corresponds to the different causalities imposed on each of these resistors. The motor gain k_m appears twice in the model; if different numerical values were inadvertently assigned to each instance, the model would no longer obey physical laws. (c) An alternative motor model shows how diagram (b) would need to be modified to cause the model to output shaft speed in response to an applied load, a change that might be required to permit unit testing of the motor submodel. Note that, as well as reversing some arrow directions, the integrator $1/s$ corresponding to the motor inertia (red) must be replaced with a differentiator s, and the sign of the signal from the motor torque to the summing junction must be reversed. (d) The gearbox block diagram includes the gearing, gear inertia, and friction. Note that the gear ratio n (brown) appears twice in the model; once again, different values assigned to each block result in a nonphysical system. (e) The flexible-beam block diagram comprises the beam compliance, inertia, and linear friction model. Note that the output torque is calculated by the beam compliance (green), which cannot therefore be easily removed.

Simple rules govern the transformation of bond graphs into other system representations.

stable controllers. One example of physical-model-based control is the impedance control methodology of [7], which is applied in the robotics field and, more recently, in the design of structural dynamics experiments [8].

Perhaps the simplest version of physical-model-based control is the realization that a PID controller is analogous to a mass-spring-damper system and can therefore (as in Case Study 1) be treated as a physical system attached to the controlled physical system. Using bond-graph representations, Figure 11 shows the physical systems corresponding to two versions of PID control. In particular, if $k_p = k$, $k_i = (k/T_i)$, and $k_d = kT_d$, then Figure 11(a) corresponds to the PID controller

FIGURE 11 PID Control. (a) Pure PID. $\mathbf{R}:\mathbf{k_p}$, $\mathbf{C}:\mathbf{k_i}$, and $\mathbf{I}:\mathbf{k_d}$ give the proportional, integral, and derivative terms, respectively; $\mathbf{I}:\mathbf{k_d}$ has *derivative* causality. (b) Filtered PID. $\mathbf{R}:\mathbf{k_f}$ and the associated **1** junction give a lowpass filtering effect on the derivative action; $\mathbf{I}:\mathbf{k_d}$ now has *integral* causality.

$$u = k\left(1 + \frac{1}{T_i s} + T_d s\right)(w - y).$$

It is well known that such a control, although a convenient idealization, is not practical due to the pure derivative action, which is reflected in the fact that the **I** component in Figure 11(a) must be assigned derivative causality. Figure 11(b) shows the bond graph corresponding to a PID control with filtered derivative. Note that the **I** component now has integral causality. In particular if $k_f = (T_f/kT_d)$, the controller equation becomes

$$u = k\left(1 + \frac{1}{T_i s} + T_d \frac{s}{1 + T_f s}\right)(w - y).$$

A crucial feature of this approach is that the system input u and output y must be colocated, that is, the bond at the lower right of each controller of Figure 11 can be directly connected to a port of the controlled system. If this is not the case and u and y are associated with different system ports, one approach is to revert to conventional control-system design based on the state-space approach or the block-diagram approach. In other words, the bond-graph methodology is used just for system modeling, and the bond graph is converted into state-space or transfer function form to be analyzed using conventional software tools such as Matlab or Octave. As an alternative, current research investigates physical-model-based control in the noncolocated case [8], [9].

Inversion and Bicausality

The systems of Figure 4 have the acausal bond graph of Figure 4(d). When natural causality is imposed as in Figure 7, the system input u and output y are

$$u = \begin{bmatrix} f_0 \\ e_1 \end{bmatrix}, \quad y = \begin{bmatrix} e_0 \\ f_1 \end{bmatrix}. \tag{21}$$

The corresponding system transfer function is

$$G = \begin{bmatrix} \frac{m_1 s + r_1}{c_1 m_1 s^2 + c_1 r_1 s + 1} & \frac{1}{c_1 m_1 s^2 + c_1 r_1 s + 1} \\ \frac{1}{c_1 m_1 s^2 + c_1 r_1 s + 1} & \frac{-c_1 s}{c_1 m_1 s^2 + c_1 r_1 s + 1} \end{bmatrix}. \tag{22}$$

This system has two poles corresponding to the components $\mathbf{C}:\mathbf{c_1}$ and $\mathbf{I}:\mathbf{m_1}$ in integral causality.

For some applications, such as actuator sizing [10], [11], it is of interest to compute inputs in terms of outputs, which leads to system inversion. Because of their acausal nature, [such as Figure 4(d)], bond-graph models are amenable to inversion. In Figure 12(a) the causality at the left-hand port is reversed to change the choice of system input and output from (21) to

$$u = \begin{bmatrix} e_0 \\ e_1 \end{bmatrix}, \quad y = \begin{bmatrix} f_0 \\ f_1 \end{bmatrix}. \tag{23}$$

The transfer function arising from this choice is

$$G = \begin{bmatrix} \frac{c_1 m_1 s^2 + c_1 r_1 s + 1}{m_1 s + r_1} & \frac{-1}{m_1 s + r_1} \\ \frac{1}{m_1 s + r_1} & \frac{-1}{m_1 s + r_1} \end{bmatrix}. \tag{24}$$

Not surprisingly, this transfer function is improper; the fact that it has only one pole follows from the fact that only the component $\mathbf{I}:\mathbf{m_1}$ is in integral causality.

Another choice of inputs and outputs is

$$u = \begin{bmatrix} e_1 \\ f_1 \end{bmatrix}, \quad y = \begin{bmatrix} e_o \\ f_o \end{bmatrix}. \tag{25}$$

However, the concept of bond-graph causality must be extended in this case. Remember that each bond carries two covariables, namely effort and flow. The placement of a single causal stroke at one end of the bond, as in Figure 6, indicates which component sets the effort and which sets the flow. This configuration is unicausal. By contrast, a bicausal model contains bonds in which one component sets both the effort and the flow. In Figure 12(b), bicausality is signified on a bond graph by separating the causal stroke into two half strokes [12], [13].

The model corresponding to the bond graph of Figure 4(d) can now be inverted by changing the causality of the bonds to that depicted in Figure 12(c). The resulting improper transfer function is

$$G = \begin{bmatrix} 1 & m_1 s + r_1 \\ c_1 s & c_1 m_1 s^2 + c_1 r_1 s + 1 \end{bmatrix}. \tag{26}$$

The transfer function has no poles because both $\mathbf{C}:\mathbf{c_1}$ and $\mathbf{I}:\mathbf{m_1}$ are now in derivative causality [Figure 6(b)].

It is worth emphasizing the qualitative aspects of inversion by this method. Model inversion is accomplished by changing the causality at the model interfaces. Changes in causality propagate through the model automatically and thus require no change to the model itself. Although transfer functions are given for completeness, the number of poles and zeros is deduced by counting the number of components in integral causality in the bond graphs of the system and inverse system, respectively.

FIGURE 12 Bicausality and Inversion. (a) Partial inversion. The RCI model of Figure 7 can be partially inverted by changing the causality at the left-hand system port. The output f_0 of the model is the flow required to cause the effort e_0 to track a desired signal given a disturbance e_1. (b) Bicausality. Using half strokes, each bond can have one of four bicausal configurations. (c) Inversion. The RCI model of Figure 7 can be inverted by changing the causality at the system interfaces. These changes propagate through the model and cause $\mathbf{C}:\mathbf{c_1}$ and $\mathbf{I}:\mathbf{m_1}$ to be placed in derivative causality. The output of the model f_0 is the flow required to cause the flow f_1 to track a desired signal given a disturbance e_1; e_0 is the corresponding effort, and $e_0 f_0$ is the power required.

Hierarchical Systems

Simple bond graphs can be constructed entirely from the standard basic components listed in Table 2. These models can be used in further bond graphs to create models of greater complexity, namely, hierarchical bond-graph models. Hierarchical bond graphs offer many of the benefits that are frequently associated with object-oriented programming techniques.

When working with bond graphs, it is natural to construct models by performing a top-down decomposition of the system of interest. By encapsulating low-level functionality within self-contained component models, clutter can be minimized easing visual inspection, thus helping the modeler focus on an appropriate level of abstraction at each stage of the model development. Such decomposition can also greatly ease the process of verifying that the structure of

The bond-graph method is a useful modeling tool, particularly well suited for describing physical systems.

the model accurately mimics the structure of the system that it is intended to represent. Subsystem encapsulation makes it easy to maintain libraries of components and swap entire classes of components within a model. Therefore, it is appropriate in the early stages of model construction to use very simple subsystem models. As model development progresses, components can be refined, and subsystems of greater complexity can be constructed, tested, and inserted into the higher-level model.

Hierarchical models are implemented using external port components, which are equivalent to combined source-sensors. These external ports are connected to other subsystem models using standard component bonds, with causality assigned in the same way as for any other bond. It is therefore possible for the causality of the bonds at the external interfaces to change. This property is particularly useful for embedding bond-graph models within system-of-systems models, where the subsystem models adapt their causal configuration to the context in which they find themselves. This feature of external ports without fixed causality is a significant advantage of bond graphs over block-diagram-based modeling methods.

Hybrid Systems

Many useful engineering systems incorporate switches, that is, components that fundamentally change the global nature of the system by making or breaking local connections. The effects of switches on the global system are generally much larger than the local physical effects of the switching action itself, and thus it is often useful to represent switches as an instantaneous change in a system variable.

When the switch establishes a connection within a system, any causal configuration can be meaningful across the switch. However, when it is used to break a connection, there is generally a definite causal configuration associated with it. Opening an electrical contact or closing a fluid valve can be represented on a bond graph by imposing zero flow on the **1** junction representing the wire or pipe in which the switch or valve is placed. Similarly, the imposition of zero effort on a **0** junction models the effect of breaking a mechanical link.

In certain circumstances, it is desirable to ensure that the inclusion of a switch does not change the global causality of a model. For example, it is often desired that all **C** and **I** components be assigned integral causality so that no algebraic equations need to be solved. Inertial switches (**ISW**) or capacitive switches (**CSW**), which combine an ideal switching element with an **I** or **C** component, can be used to ensure that there is no change in causality due to the action of the switch.
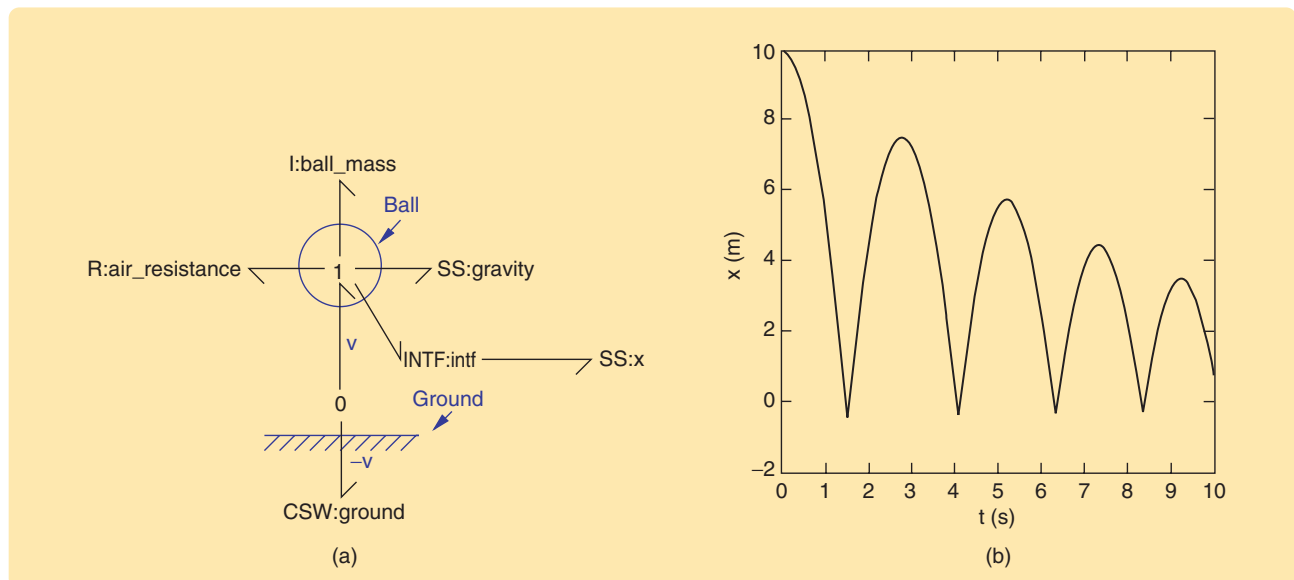
FIGURE 13 Bouncing ball. (a) The bond graph superimposed on the schematic diagram. The **CSW** component models the ground, while the **INTF** component integrates the flow (velocity) to give height. (b) In this simulation, the ball is dropped from 10 m. The air resistance reduces each rebound, while the ground compliance allows negative height.

> **Power is the universal currency of physical systems.**

Figure 13 gives an example of using a **CSW** component in a mechanical context. When the ball is above ground, the **CSW** is open and no force is exerted on the ball; when the ball is below ground, the **CSW** acts as an ordinary spring, thus creating a bounce.

Distributed-Parameter Systems

Although the bond-graph approach does not explicitly handle distributed-parameter systems described by partial differential equations, these systems can be approximated using N discrete lumps. For example, the flexible beam of Figure S1(a), approximated in S1(c) by a single lump, can be better approximated using N lumped elements each of the form of Figure 14(a). Using the Bernoulli-Euler approximation, each lump has the bond-graph representation of Figure 14(b).

Figure 14(c) shows the frequency response relating angle and angular velocity at the fixed end of the beam for different N. The final choice of N depends on the bandwidth over which the approximation is required.

CASE STUDY 2: AN AIRCRAFT FUEL SYSTEM

The second case study is an aircraft fuel system modeled using bond graphs. The design presented is a hybrid of three aircraft.
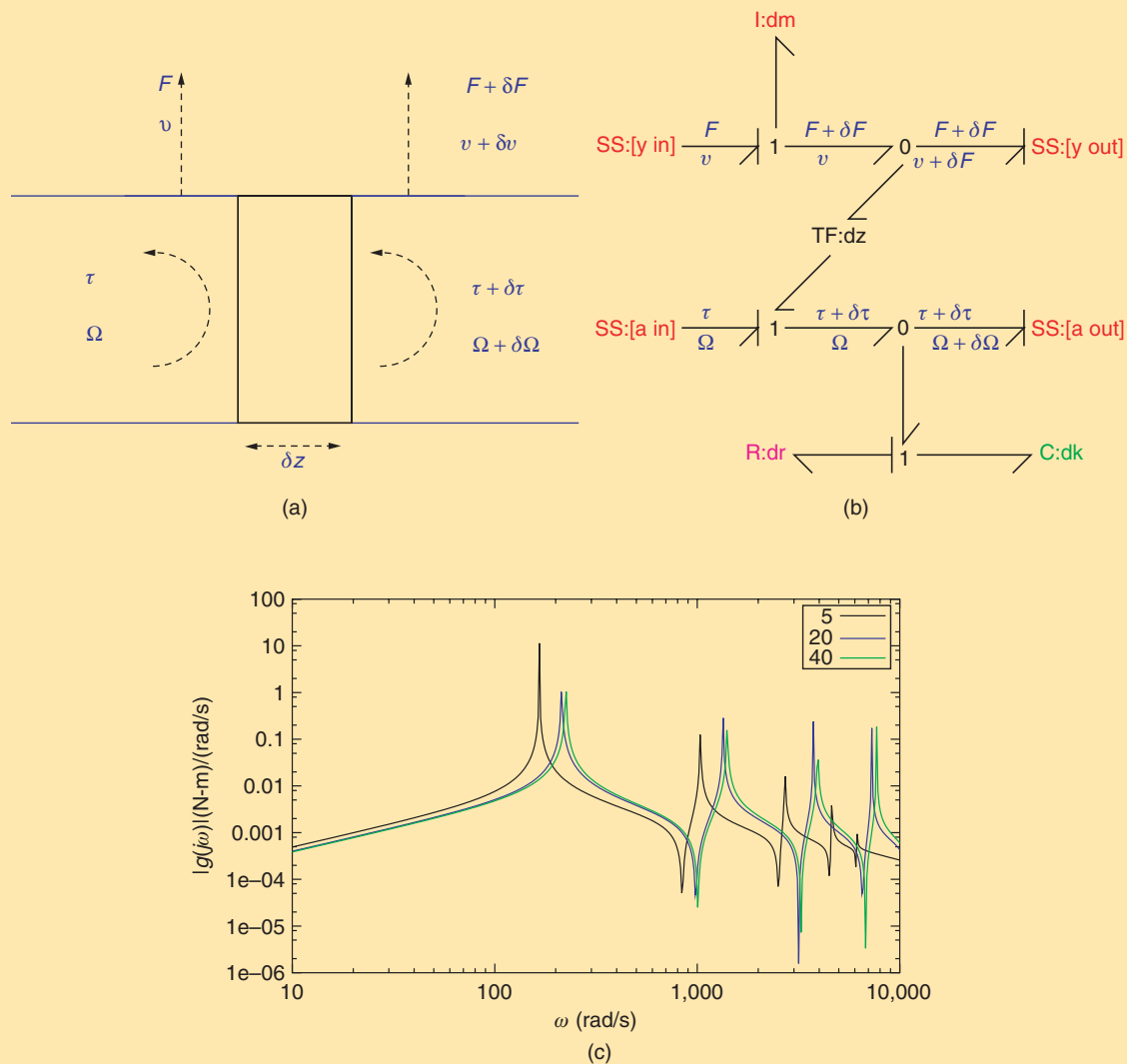
FIGURE 14 A uniform beam. (a) An infinitesimal lump of width δz. The vertical velocity v is driven by the net force δF, while the torque τ is driven by the net angular velocity $\delta\Omega$. (b) Lumped bond graph. The interaction between angular and linear motion is expressed by the transformer **TF:dz**, the linear motion of the lump is expressed by **I:dm**, and the angular twist by **C:dk**. The term **R:dr** expresses structural damping. (c) Frequency responses $|g(j\omega)|$ for N lumped elements, where $N = 5, 20, 40$.

The model that we develop is suitable for preliminary fuel-system architectural design, typically performed during the conceptual design phase, or as a realistic fuel system substitute for integration testing during later design phases. More interestingly perhaps for control engineers, the model is eminently suitable for use in specifying, developing, and verifying complex system control software, which is the most expensive part of most modern aircraft development.

For architectural design work, the main requirements are that the model should be easy to modify, easy to analyze, and give reasonably accurate results for minimal effort if the design changes frequently and questions about
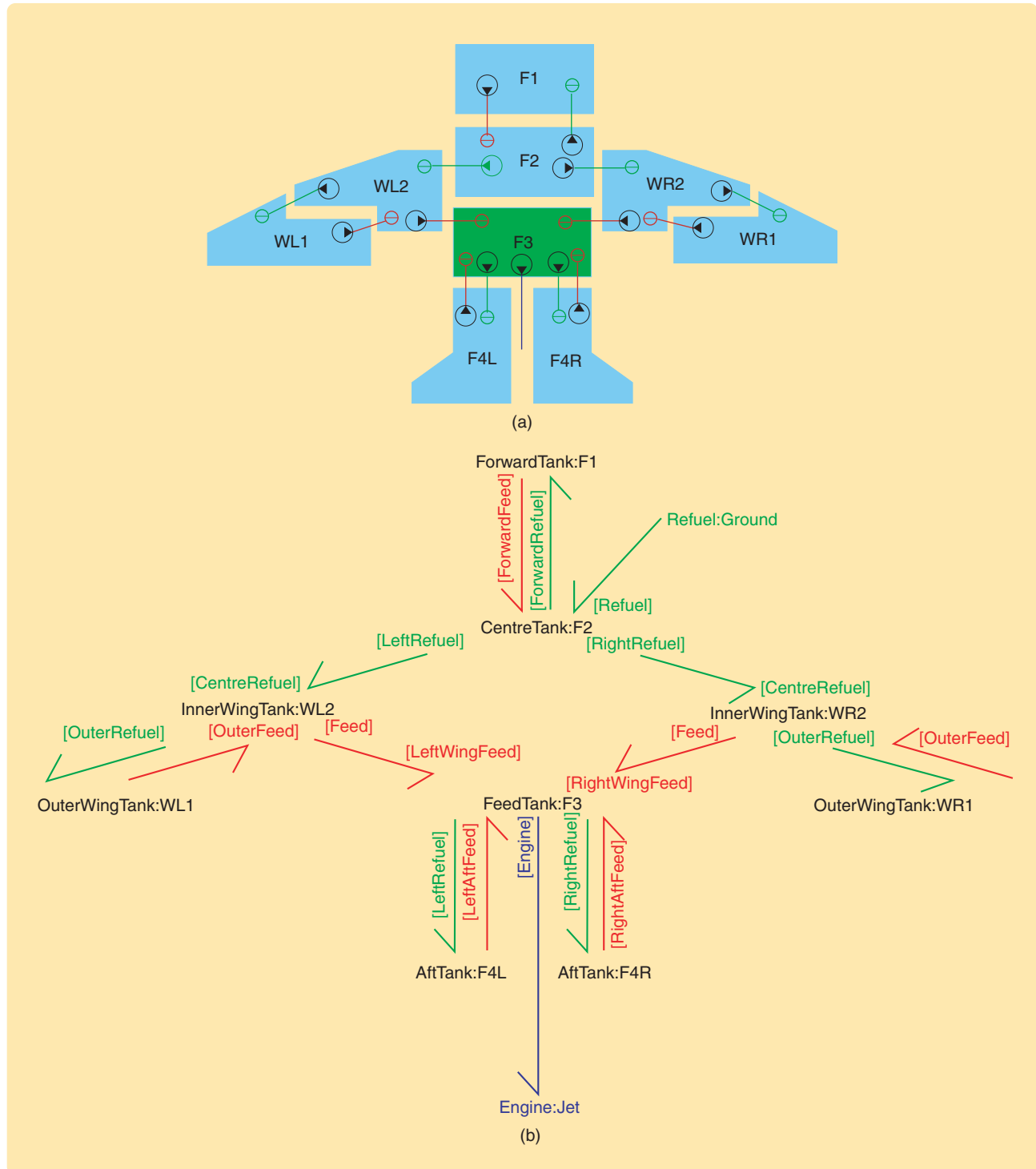
FIGURE 15 A simple aircraft fuel model. (a) Fuel tank layout. (b) Bond graph. The refuel system is depicted in green; the feed system in red. The system consists of nine tanks of which there are six types.
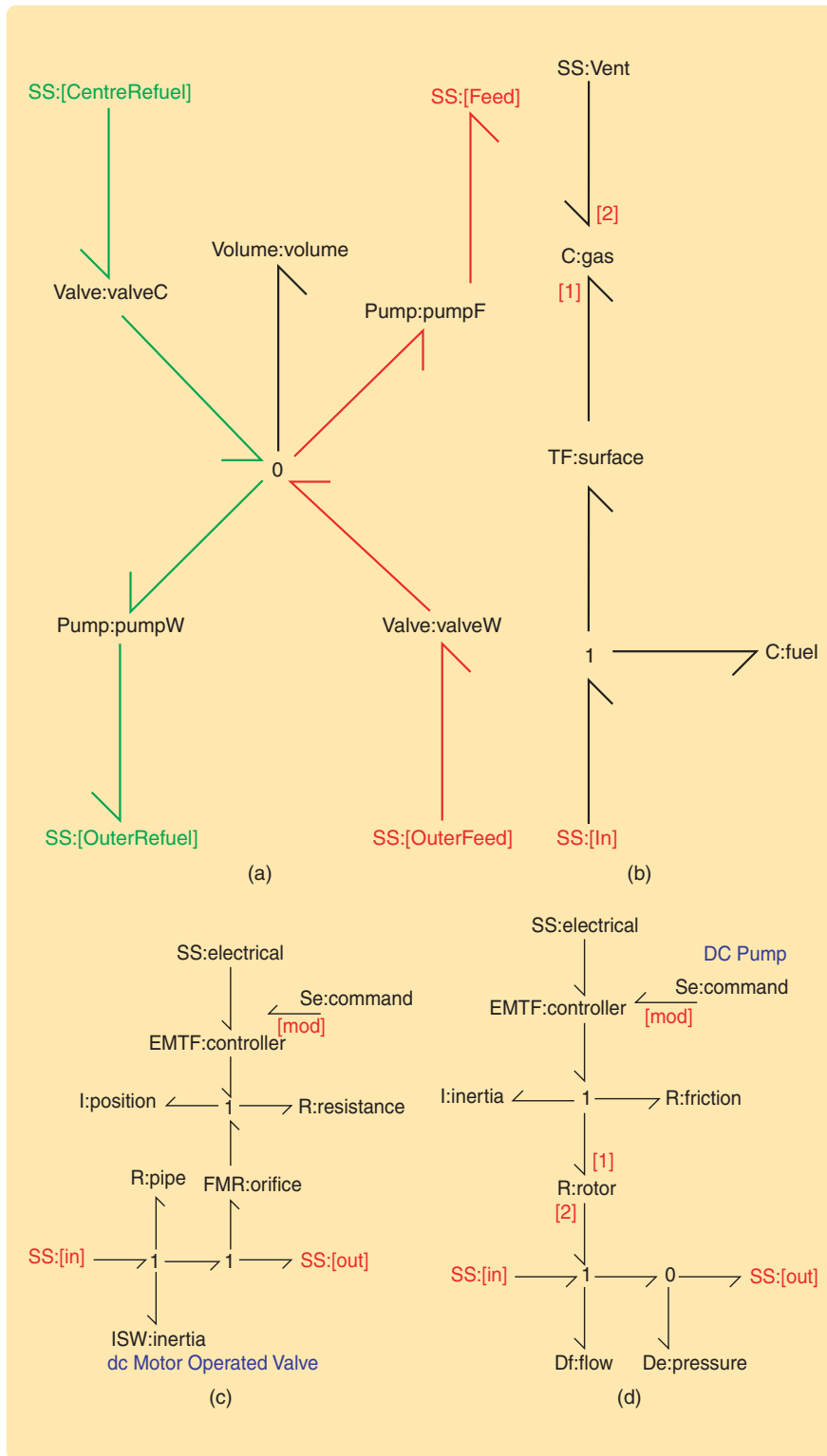
FIGURE 16 Submodels. (a) Tank. Each tank submodel includes the **Valve** and **Pump** components, which are located within the tank, and a **Volume**, which contains the fuel and inert gas. (b) **Volume**. This component represents the storage of the fuel and inert gas, which is contained within the tank. The fuel is stored in a **C** component with an incompressible constitutive relationship. The gas is stored in a **C** component with a compressible constitutive relationship. A **TF** represents the surface separating the two fluid domains. (c) **Valve**. (d) **Pump**. These components are the electromechanical actuators that control the movement of the fluid within the fuel system. Interfaces with the dc electrical system are included in the component models.

system performance must be answered quickly.

For integration testing and control software development, models must typically be capable of running in real time, while exhibiting the main features of the system behavior and dynamically exciting all relevant interfaces through which the model is connected to other systems. In these cases, models are often used by people not closely associated with the system hardware itself and so must be reliable and not require specialized knowledge of the system in order for the model to be operated successfully.

Model Design

The **Fuel** model [Figure 15(b)] is a hierarchical bond graph representing a simple aircraft fuel-management system. The system comprises nine fuel tanks, of which there are six types, namely, one instance each of the **ForwardTank**, **CentreTank**, and **FeedTank** fuel tanks, and the **InnerWingTank** [Figure 16(a)], **OuterWingTank** and **AftTank** types, which are each instantiated twice due to the lateral symmetry of the aircraft. For simplicity, the system is athermal, and the pipework is assumed to be coplanar and perpendicular to the external gravitational field. The connections between the tanks form two major subsystems: Refuel (green) and Feed (red) the respective purposes of which are to load fuel into the aircraft and transfer it from the storage tanks to the feed tank, which supplies the jet engine. These subsystems can also be used to transfer fuel around the aircraft according to the demands of the flight control system.

Fuel enters the system through the ground refueling point (**Refuel:Ground**) attached to the center fuselage tank (**CentreTank:F2**) and leaves the system through the single jet engine (**Engine:jet**). The engine is supplied with fuel from the feed tank (**FeedTank:F3**).

Each of the six tank types contains a **Volume** [Figure 16(b)], which contains stored fuel and inert gas, and **Valve** and **Pump** components [Figure 16(c) and (d)], which allow fuel to enter or leave the tank.

The valves contain resistive losses representing the valve orifices and the pipes to which they are connected. The frictional losses in the pipes are implemented with a simple resistive component (**R:pipe**). Losses associated with the valve orifice are implemented using a flow-modulated resistor (**FMR:orifice**), the resistance of which is modulated according to the valve position, which is controlled by an electronic actuator. Inertial switches are used to allow or disallow the flow of fuel through the valves.

The **Valve** and **Pump** components contain controllers that regulate the flow of fuel. The control logic is implemented as a simple text file, which actuates these components according to the quantity of fuel in the tanks (the system states). The simple aim of the control logic as implemented is to keep the feed tank as full as possible at all times so that the engine does not run dry. The results of a simulation using this model and the control logic can be seen in Figure 17, which shows the volume levels of fuel in the tanks as fuel is consumed by the engine. A real fuel-management system would have additional requirements, such as apportioning the remaining fuel between the other tanks to control the center of gravity of the fuel to within appropriate longitudinal and lateral limits for the flight mode to enhance agility, or maintain stability while optimizing range.

This article is an introduction to bond graphs.

Benefits of Modeling the System as a Bond Graph

Implementing the fuel system model as a bond graph provides several benefits over traditional signal flow models that would typically be produced using tools such as Simulink or EASY5. Perhaps the most important benefit is the adaptable nature of the interfaces. After development and testing, component and subsystem models must be capable of being embedded in diverse environments and able to adapt to a range of conditions.

Consider the refuel system. Design of pipe geometry requires that pressure drops remain within certain limits when maximum flow is forced through the pipes, so the model must be able to accept a flow of fuel as an input at the refueling end, and the pipes must determine the system pressure loss. However, to verify that the resulting design meets performance requirements relating to maximum permissible refuel time, it is necessary to instead perform simulations with an appropriate fuel pressure applied at the input and fuel flow rates calculated through the pipes and valves. Signal flow models would require separate models, with a consequent doubling of the amount of testing required.

The need for additional models would also mean that more new models are used throughout the design process, with a risk of new errors being introduced. It is generally preferable to use existing tried-and-tested models wherever possible.

Adaptable interfaces are also beneficial when integrating system models to produce a virtual integrated aircraft model. If all of the systems are produced as bond graphs, all of the interfaces are guaranteed to transact power, which greatly simplifies the task of stitching models together. The task is further simplified by the fact that the causality of the interfaces can be changed without producing new models. Thus, an electric fuel pump, which accepts a constant voltage input during fuel system simulations, can easily be adapted to output
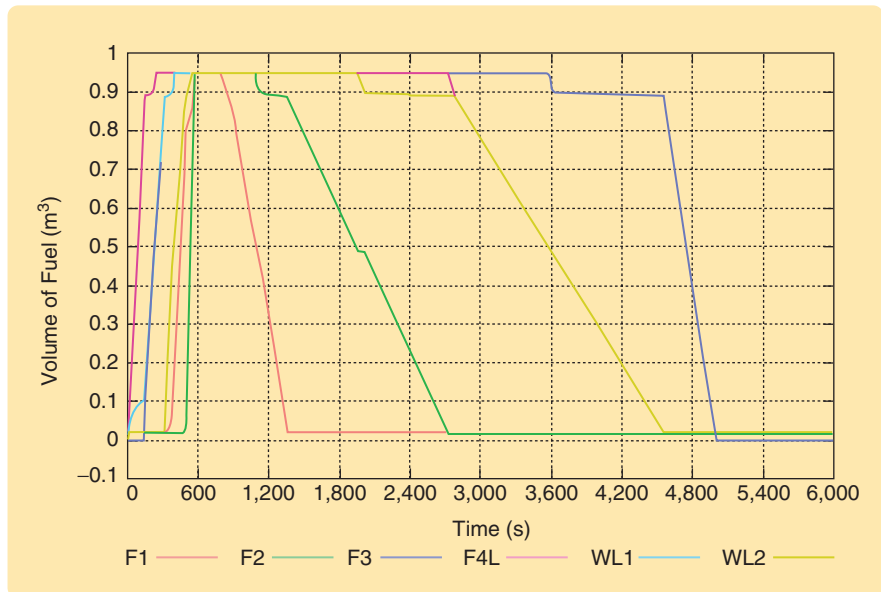
FIGURE 17 Simulation of the hybrid fuel model. The volume of fluid in each tank changes as fuel is consumed by the jet engine. A simple logic controller aims to keep the feed tank (F3) replenished from the transfer tanks, using the valves and pumps. The steps at 0.9 m^3 are due to thresholds within the control logic that prevent valve chatter. The volume of fuel in the transfer tanks never quite reaches zero because residual fuel always remains due to tank geometry; this behavior is modeled by low-level thresholds in the control logic. A more sophisticated controller would attempt to also maintain the center of gravity within acceptable limits.

Bond graphs are inherently energy based and thus are related to other energy-based methods, including dissipative systems and port-Hamiltonian systems.

a voltage when connected to a current-producing electrical distribution model.

The compact nature of bond graphs also makes them ideal for representing fluid flow, which is just a particular form of energy transport. Unlike signal flow diagrams, it is not necessary to take any particular care with the sign convention for flows at each port of a component model. Since the bond-graph sign convention is simply specified by the direction of bonds impinging on a component, there is no difficulty with removing components and embedding them in other parts of the system model.

It should be noted that producing bond-graph system models does not require that control engineers give up traditional design tools. Several bond-graph software packages can convert bond graphs to formats that can be embedded directly within Matlab and Simulink—as m-files, mex files, and S functions—and within other software packages in their native formats. Using appropriate tools for each stage of work generally yields better results than attempting to use the most readily available tool for all aspects of system design.

We recommend creating models with specialist modeling tools, running simulations within appropriate simulation harnesses, and performing control design with appropriate design tools. Pencil and paper can of course be substituted for software at any stage of the development process except real-time simulation.

Related Paradigms

Bond graphs are related to other modeling approaches. Two of these of particular interest are the behavioral approach and energy-based methods, such as dissipative systems and the port-controlled Hamiltonian approach.

The behavioral approach [15], [16] and the bond-graph approach provide two of the many ways of describing and understanding dynamical systems. As discussed in more detail in [17], the two approaches are similar in that

- the system description does not distinguish inputs and outputs, but rather is viewed as a constraint on a set of variables: the manifest variables in behavioral terms and port variables in bond-graph terms
- systems are connected without assigning the input/output structure beforehand
- state-variable descriptions are regarded as representations to be derived from the basic system representation only when decisions have been made about which variables are to be regarded as inputs and outputs.

The two approaches are different in that

- the bond-graph approach is graphical, whereas the behavioral approach is mathematical
- the bond-graph approach is explicitly based on energy concepts and uses the systematic modeling approach whereby physical system variables are classified according to Table 1
- the behavioral approach handles distributed systems, described by partial differential equations. The bond-graph approach does not.

It is becoming recognized that energy-based methods are relevant to control engineers, see [18]. Bond graphs are inherently energy based and thus are related to other energy-based methods, including dissipative systems [19]–[21] and, as shown in [22], port-Hamiltonian systems.

CONCLUSIONS

This article has presented an introduction to bond graphs for control engineers. Although the notation can initially appear daunting, the bond graph method is firmly grounded in the familiar concepts of energy and power. The essential element to be grasped is that bonds represent power transactions between components. Engineers can quickly become adept at applying the technique with just a little practice.

The use of generic components and variables makes it a simple matter to model multidomain systems, allowing engineers to analyze complex problems and interactions that might normally be hidden by more traditional approaches to subsystem division and analysis.

The graphical nature of bond graphs enables modelers to easily identify potentially troublesome areas of their system representations and to quickly determine the form of remedy that can make the model more appropriate for the task in hand. The method is particularly beneficial in identifying where supposedly simplifying assumptions and approximations might be counterproductive.

Simple rules govern the transformation of bond graphs into other system representations, and readily available software exists to perform conversions automatically. Systems modeled as bond graphs can thus be easily integrated with familiar control engineering toolsets.

The bond-graph approach is not unique in focusing on energy and in determining causality after modeling. Alternative approaches with these characteristics are discussed in "Related Paradigms." However the bond-graph approach is unique in combining these features with an intuitively appealing graphical modeling and causality analysis formulation.

We believe that the bond-graph method is a useful modeling tool, particularly well suited for describing physical systems, and can provide a powerful way for engineers to analyze and solve the problems that they face.

REFERENCES

[1] H. Paynter, "An epistemic prehistory of Bond Graphs," in P. Breedveld and G. Dauphin-Tanguy, Eds., *Bond Graphs for Engineers*. Amsterdam: North-Holland, 1992, pp. 3–17.

[2] W. Marquis-Favre and S. Scavarda, "Alternative causality assignment procedures in bond graph for mechanical systems," *Trans. ASME*, *J. Dyn. Syst., Meas. Control*, vol. 124, no. 3, pp. 457–463, 2002.

[3] D. Karnopp, D.L. Margolis, and R.C. Rosenberg, *System Dynamics: Modeling and Simulation of Mechatronic Systems,* 3rd ed. New York: Horizon, 2000.

[4] Octave, "Octave home page," 2002 [Online]. Available: http://www.octave.org

[5] A. Sharon, N. Hogan, and D.E. Hardt, "Controller design in the physical domain," *J. Franklin Inst.*, vol. 328, no. 5, pp. 697–721, 1991.

[6] P.J. Gawthrop, "Physical model-based control: A bond graph approach," *J. Franklin Inst.*, vol. 332B, no. 3, pp. 285–305, 1995.

[7] N. Hogan, "Impedance control: An approach to manipulation. Part I–Theory," *Trans. ASME*, *J. Dyn. Syst., Meas. Control*, vol. 107, pp. 1–7, Mar. 1985.

[8] P. Gawthrop, M. Wallace, and D. Wagg, "Bond-graph based substructuring of dynamical systems," *Earthquake Eng. Struc. Dyn.*, vol. 34, pp. 687–703, May 2005.

[9] P.J. Gawthrop, "Bond graph based control using virtual actuators," *J. Syst. Control Eng.*, vol. 218, pp. 251–268, Sept. 2004.

[10] R.F. Ngwompo and P.J. Gawthrop, "Bond graph based simulation of nonlinear inverse systems using physical performance specifications," *J. Franklin Inst.*, vol. 336, pp. 1225–1247, Nov. 1999.

[11] R.F. Ngwompo, S. Scavarda, and D. Thomasset, "Physical model-based inversion in control systems design using bond graph representation Part 1: theory," *J. Syst. Control Eng.*, vol. 215, pp. 95–103, Apr. 2001.

[12] P.J. Gawthrop, "Bicausal bond graphs," in *Proc. Int. Conf. Bond Graph Modeling Simulation (ICBGM'95)*, Las Vegas, 1995, vol. 27, pp. 83–88.

[13] P.J. Gawthrop, "Physical interpretation of inverse dynamics using bicausal bond graphs," *J. Franklin Inst.*, vol. 337, pp. 743–769, 2000.

[14] J. Apkarian, *A Comprehensive and Modular Laboratory for Control Systems Design and Implementation*. Quanser Consulting, 1995.

[15] J.W. Polderman and J.C. Willems, *Introduction to Mathematical System Theory: A Behavioral Approach* (Applied Mathematics Series 26). New York: Springer-Verlag, 1997.

[16] J. Willems, "On interconnections, control, and feedback," *IEEE Trans. Automat. Contr.*, vol. 42, pp. 326–339, Mar. 1997.

[17] P.J. Gawthrop, "Bond graphs in a behavioral context," in *Proc. 13th European Simulation Symp.: Simulation in Industry*, Marseille, France, 2001, pp. 745–749.

[18] R. Ortega, A.J. van der Schaft, I. Mareels, and B. Maschke, "Putting energy back in control," *IEEE Control Syst. Mag.*, vol. 21, pp. 18–33, Apr. 2001.

[19] J.C. Willems, "Dissipative dynamical systems, Part I: general theory, Part II: Linear system with quadratic supply rates," *Arch. Rational Mechanics Anal.*, vol. 45, no. 5, pp. 321–392, 1972.

[20] D.J. Hill and P.J. Moylan, "Dissipative dynamical systems: Basic input-output and state properties," *J. Franklin Inst.*, vol. 208, no. 5, pp. 327–357, 1980.

[21] R. Lozano, B. Brogliato, O. Egelund, and B. Maschke, *Dissipative Systems: Analysis and Control*. New York: Springer-Verlag, 2000.

[22] D. Vink, D. Ballance, and P. Gawthrop, "Bond graphs in model matching control," *Math. Comput. Model. Dyn. Syst.*, vol. 12, no. 2-3, pp. 249–261, 2006.

[23] A. Mukherjee, R. Karmakar, and A. Samantaray, *Bond Graph in Modeling, Simulation and Fault Detection*. New Delhi: I.K. International Publishing, 2006.

[24] P.J. Gawthrop and L.P.S. Smith, *Metamodelling: Bond Graphs and Dynamic Systems*. Hemel Hempstead, Herts, England: Prentice Hall, 1996.

[25] P. Breedveld and G. Dauphin-Tanguy, Eds. *Bond Graphs for Engineers*, Amsterdam: North-Holland, 1992.

[26] F.E. Cellier, *Continuous System Modelling*. New York: Springer-Verlag, 1991.

[27] F.E. Cellier and J.J. Granda, in *Proc. 1995 Int. Conf. Bond Graph Modeling Simulation (ICBGM'95)*, Las Vegas, 1995, vol. 27.

[28] J.J. Granda and F.E. Cellier, in *Proc. Int. Conf. Bond Graph Modeling (ICBGM'93)*, La Jolla, CA, 1993, vol. 25.

[29] J.J. Granda and G. Dauphin-Tanguy, in *Proc. 1997 Int. Conf. Bond Graph Modeling Simulation (ICBGM'97*, Phoenix, AZ, 1997, vol. 29.

[30] J.J. Granda and F. Cellier, in *Proc. 1999 Int. Conf. Bond Graph Modeling Simulation (ICBGM'99)*, San Francisco, CA, 1999, vol. 31.

[31] J.J. Granda and F. Cellier, in *Proc. Int. Conf. Bond Graph Modeling Simulation (ICBGM'01)*, Phoenix, AZ, 2001, vol. 33.

[32] *Proc. Int. Conf. Bond Graph Modeling and Simulation (ICBGM'03)*, Simulation Series, Orlando, Florida, U.S.A.: Society for Computer Simulation, 2003.

[33] *Proc. Int. Conf. Bond Graph Modeling and Simulation (ICBGM'05)*, Simulation Series, New Orleans, U.S.A.: Society for Computer Simulation, 2005.

[34] P.C. Breedveld, "Editorial: Special issue on bond graphs," *J. Franklin Inst.*, vol. 328, no. 5-6, pp. 523–524, 1991.

[35] P.J. Gawthrop and S. Scavarda, "Special Issue on Bond Graphs: Editorial," in *Proc. Institution of Mechanical Engineers Pt. I: J. Systems and Control Engineering*, Mar. 2002, vol. 216, pp. i–v.

[36] W. Borutzky and P. Gawthrop, "Special issue on bond graph modelling (Editorial)," *Math. Comput. Model. Dyn. Syst.*, vol. 12, pp. 103–105, Apr.-June 2006.

AUTHOR INFORMATION

Peter J. Gawthrop (p.gawthrop@mech.gla.ac.uk) obtained his B.A., D.Phil., and M.A. degrees in engineering science from Oxford University in 1973, 1977, and 1979, respectively. Following a period as a research assistant with the Department of Engineering Science at Oxford University, he became W.W. Spooner Research Fellow at New College, Oxford. He then moved to the University of Sussex as a lecturer, and later a reader in control engineering. In 1987, he took up the Wylie Chair of Control Engineering in the Department of Mechanical Engineering at Glasgow University, where he was a founding member of the Centre for Systems and Control. Since 2002, he has held an honorary research position at Glasgow. He has also held visiting appointments in Australia at the universities of New South Wales, Newcastle, Sydney, and the Royal Melbourne Institute of Technology. In the United Kingdom, he has a visiting position at the University of Bristol. His research interests include continuous-time model-based predictive control, intermittent control, physiological control systems, system identification, and system modeling and control using the bond-graph method. He can be contacted at the Centre for Systems and Control, University of Glasgow, Glasgow G12 8QQ, U.K.

Geraint P. Bevan graduated from the University of Salford in 1996 with a bachelor of engineering degree in aeronautical engineering. He then joined British Aerospace and worked on several aircraft projects in the United Kingdom and the United States until 2003, when he became a research assistant at the University of Glasgow, where he is also a part-time doctoral student. He is currently undertaking research to develop automatic automotive controllers for active vehicle safety within the CEmACS project, part of the European Sixth Framework Programme.